Security and Privacy in Cloud-Enabled Cyber-Physical Systems

Amir Mohammad Naseri

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Information System Security) at

Concordia University

Montréal, Québec, Canada

December 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Amir Mohammad Naseri**

Entitled: **Security and Privacy in Cloud-Enabled Cyber-Physical Systems**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Information System Security)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Amr Youssef

______________________________ Examiner
Dr. Amr Youssef

______________________________ Examiner
Dr. Mohammad Mannan

______________________________ Supervisor
Dr. Walter Lucia

Approved by ______________________________
Dr. Zachary Patterson, Graduate Program Director

__________ 2022

____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Security and Privacy in Cloud-Enabled Cyber-Physical Systems

Amir Mohammad Naseri

The advent of Cyber-Physical Systems (CPS)s is considered a revolution in the industry's modern history. CPSs are anticipated to have a rapid diffusion in safety-critical domains such as intelligent transportation, energy distribution, and industry 4.0. Control systems are the core of any CPS since they are in charge of deciding the control inputs given the measurements provided by distributed sensors. Advanced control algorithms require a significant amount of computational power that might not be available on-site. In these scenarios, cloud computing represents a possible solution. Ensuring the cyber-security of cloud-enabled CPSs is an important concern, especially when they are used in safety-critical applications. Indeed, a malicious cloud provider can misuse the sensor measurements and/or control inputs or sabotage the control algorithm.
In this thesis, we investigate different security and privacy issues in cloud-based control systems and provide different control-theoretical solutions to enhance their cyber security.

By assuming a cloud-based CPS, we show three different approaches to ensure the privacy of the controller operations, sensor measurements, and control inputs. In particular, we propose solutions based on (i) an outsourced transformed control problem, (ii) an encrypted control strategy, and (iii) a trusted execution environment. While the first two approaches are effective against passive attackers, the third one is effective also against active ones.

Then, we consider networked control systems where the controller operations are implemented on encrypted data exploiting homomorphic cryptosystems. In this setup, we show that an active attacker with access to the control logic in the cloud can exploit the small domain of the message space and the randomization process required to make the utilized ciphers semantically secure to break the secrecy of the cryptosystem and/or establish a covert channel between the cloud and an eavesdropper on the measurement channel.

Finally, we address the problem of establishing a secret key between the plant and a remote controller without resorting to traditional cryptographic techniques. By considering, as case of study, a remotely controlled mobile robot, we show that an observer-based protocol can be used to securely agree on a secret key. The validity of the proposed solution has been tested on a laboratory robot.

Acknowledgments

I would like to express my gratitude and sincere appreciation to my supervisor Dr. Walter Lucia for his immense help and support throughout my master's degree. This thesis and all my other achievements during the master's would not be possible without his continuous guidance and insightful comments. I cannot be more thankful to have such a knowledgeable, considerate, supportive, and outstanding supervisor who convincingly guided me to move in the right and professional direction. I am very proud and happy that I had the opportunity to spend more than two years of my academic life with him.

Also, I would like to express my deepest gratitude to Dr. Amr Youssef, whose sincerity and encouragement I will never forget. I am thankful for the extraordinary experiences he arranged for me and for providing opportunities for me to grow professionally. It is an honor to learn from Dr. Youssef.

Last but not least, nobody has been more important to me in the pursuit of this thesis than my family. I want to thank my wife, Sharareh Younesian, who none of this would have been possible without her encouragement and help. I must express my heartfelt appreciation to my beloved parents, and my little sister.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of CPSs is considered a revolution in the industry's modern history. CPSs are anticipated to have a rapid diffusion in safety-critical domains such as intelligent transportation, energy distribution, and industry 4.0. Therefore, their security against cyber-attacks is a primary concern. On the other hand, control engineering requirements are changing, and more computational power has been required to keep up with the industry demands. Cloud services can provide high computational power for different control systems. The use of such services saves on the cost of setting up and maintaining industrial control systems (ICSs), resolving issues of resource-constrained CPS, as well as off-loading computationally expensive tasks. As discussed in [Givehchi et al., 2014, Darup et al., 2021], the use of a cloud-based controller is expected to improve the economic efficiency of industrial control systems. Moreover, when ICSs are geographically distributed, these cloud services are highly available and accessible from different locations [Mahmoud and Xia, 2019]. Cloud-based intelligent transportation system is an example of such an application where the computations on the cloud can be used for different purposes such as road safety, transport productivity, travel reliability, informed travel choices, environment protection, and traffic resilience [Bitam and Mellouk, 2012]. Despite all of these advantages and many others, utilizing Cloud-Enabled Networked Control Systems (CE-NCS) increases the attack surface for the adversaries and puts the privacy of the underlying CPS in danger. So, the concern of security and privacy of CPSs will be raised.

A cloud provider should not always be considered as a fully-trusted party; it is essential to ensure the confidentiality of the information against honest-but-curious cloud providers, i.e., providers that do not deviate from the defined protocol but attempt to learn all possible information from legitimately received messages. Those cloud providers can eavesdrop on the provided information and misuse them. Moreover, the privacy of the communications channels between the plant and the cloud server is another concern that should be addressed.

1.1 Literature Review

Different approaches have been proposed to enhance the security and privacy of networked CPSs where the controller is hosted in a cloud infrastructure. In what follows, by focusing mainly on control-theoretical solutions, the state-of-art solutions are reviewed.

Differential privacy techniques were proposed for use in cloud-enabled networked control systems, e.g., see [Cortés et al., 2016]. However, these solutions require the addition of noise on top of the sensor measurements and control signals which affects the control system performance. As a result, privacy is achieved at the cost of degradation in the performance of the system. Transformation-based methods have been also utilized for privacy-preserving in cloud-enabled networked control systems. In transformation-based methods [Wang et al., 2011, Weeraddana and Fischione, 2017], the main idea is to send a transformed optimization problem to the cloud so that the solution of the original problem can be recovered from the solution of the transformed problem computed on the cloud. Xu et al. [Xu and Zhu, 2015] proposed a privacy-preserving solution for an MPC CE-NCS based on problem transformation and game-theoretic approaches.

Zhou et al. [Zhou et al., 2013] proposed the use of conventional cryptographic algorithms to secure plant-to-cloud communication for the first time. However, such a solution is required to decrypt the data on the cloud before the control algorithm can be executed. Besides the delay caused by these operations, the encryption key needs to be stored on the controller side. If the stored key is leaked to the attacker by any means, the utilized cryptosystem will break. To address such an issue, homomorphic encryption (HE) has been proposed by [Kogiso and Fujita, 2015]. Generally speaking, homomorphic cryptosystems allow arithmetic operations on encrypted data. Therefore, they enable the possibility of implementing the control logic directly on the encrypted data. Such a solution has the advantage of eliminating the necessity of storing the decryption key in the cloud, and it prevents the leakage of private information (e.g., sensor measurements and control input) transmitted between the plant and the cloud. The above-described idea can be used in principle to implement different control strategies in an encrypted fashion [Darup et al., 2021]. In the literature, solutions have been proposed to implement static state-feedback controllers, dynamic output feedback regulators [Tran et al., 2020], and advanced optimal solutions such as Model Predictive Control (MPC) [Darup et al., 2018]. Of particular interest for this paper are encrypted model MPC solutions [Rawlings et al., 2017, Borrelli et al., 2017]. MPC is an advanced optimal control strategy that, in its general optimal formulation, can take into account plant constraints and disturbances. For such capabilities, this control paradigm has been successfully employed in different application domains [Mayne, 2014]. Recently, different attempts have been made to develop an encrypted scheme of MPC. The main challenge in adopting such a strategy is represented by the impossibility of solving the MPC optimization problems using encrypted data and a single cloud [Darup et al., 2019]. Therefore, existing approaches have tried to mitigate such an issue by performing part of the computations on the actuator/sensor processing units or adopting a multi-cloud infrastructure. In [Darup et al., 2017], an encrypted version of the explicit MPC strategy based on an offline-computed piece-wise partition of the control law [Bemporad et al., 2002] is developed under the assumption that the sensor has sufficient computational power to identify the partition to which the current state belongs. In [Schlüter and Darup, 2020, Alexandru et al., 2018], the MPC optimization problem is encrypted by means of a multi-party (multi-cloud) computational architecture. Although such solutions do not need any computations on the plant's side, they require an expensive infrastructure that might not be affordable in many control system applications. In [Darup et al., 2018], the authors have proposed an encrypted version of a standard online MPC scheme where the required optimization problem is split into two parts. In the first part, on the cloud, one iteration of the Proximal Gradient Method (PGM) [Darup et al., 2018] is implemented utilizing the used homomorphic cryptosystem. In the second part, in plaintext and on the actuator, the solution computed on the cloud is projected into the admissible constraint set. Given the limited computational resources typically available on the actuator, such a solution has been developed assuming that the projection can be efficiently performed, e.g., when the plant is subject to box-like input constraints. Another possible drawback of such a solution is that the single iteration might cause unacceptable performance degradation. Along similar lines is the solution proposed in [Darup, 2020] where the use of an alternating direction method of multipliers is utilized with the aim to deal with both state and input constraints. Similar to [Darup et al., 2018], the underlying assumption is that efficient projections can be performed on the actuator. Moreover, such a solution suffers from the problem that the number of operations performed on the actuator increases with the prediction horizon. Therefore, there exists an unavoidable trade-off on the choice of the prediction horizon $N_p \geq 1$. On one side, N_p must be sufficiently large to ensure feasibility [Rawlings et al., 2017]. On the other hand, N_p should be small to have an acceptable computational load on the actuator. However, these homomorphic solutions suffer from unavoidable limitations related to the arithmetic operations allowed by the homomorphic schemes, ciphertext size explosion, and computation overhead. Solutions targeting only securing communication channels cannot protect controller logic and data against a malicious or compromised cloud provider. Ensuring the confidentiality of processed data in the cloud via encrypted control systems introduces computation and communication overheads with respect to traditional non-encrypted networked control systems. However, the extra computational load is mainly related to the operations performed inside the cloud, which, in such architectures, is assumed to have high computation capabilities. Therefore, in the presence of high-performance cloud and communication infrastructures, the resulting transmission and execution delays introduced by encrypted control systems can be minimized in order to satisfy the delay constrains ts of the underlying control system [Teranishi et al., 2019]. An interesting study about the efficiency of four different homomorphic encryption schemes can be found in [Geng and Zhao, 2019]. Moreover, experimental engineering studies about the feasibility of homomorphically encrypted control systems can be found in, e.g., [Ishikawa et al., 2016, Tran et al., 2019] where the authors have proved the feasibility of such architecture to control a DC motor [Ishikawa et al., 2016] and an inverse pendulum [Tran et al., 2019].

Although encrypted control systems can, in principle, solve the security and privacy problems of NCSs, different deception attacks against these control architectures have been proposed in [Baba et al., 2018, Kogiso, 2018, Teranishi and Kogiso, 2019, Lee et al., 2020]. In [Baba et al., 2018], by exploiting the encrypted control system sensitivity to signal and parameter falsifications, an attack detector based on a low-pass filter is proposed to detect falsified control signals and parameters. In

[Kogiso, 2018], stealthy replay attacks are investigated, and a switching private/public keys management system is proposed to prevent and detect such attacks. In [Teranishi and Kogiso, 2019], first, the authors show that any encrypted control system based on homomorphic encryption can be subject to attacks exploiting the inherent malleability of the encryption scheme (i.e., the attacker can manipulate encrypted data without the need to decrypt them). In particular, if the adversary is aware of the used homomorphic scheme, then it can change sensor measurements, control parameters, and even re-assign the poles of the closed-loop system. Then, the authors propose a QR decomposition technique to prevent malleability-based pole-assignment attacks. In [Lee et al., 2020], it is shown that if the encrypted controller is implemented resorting to an additively homomorphic encryption system, then it is possible to exploit the malleability property to launch undetectable zero-dynamics attacks. However, all the above-described homomorphic solutions suffer from unavoidable limitations related to the arithmetic operations allowed by the homomorphic schemes, ciphertext size explosion, and computation overhead. Moreover, solutions only targeting the security of communication channels cannot protect the controller logic and transmitted data against a malicious or compromised cloud provider. For data and execution security in the context of IoT and CPS applications, particularly interesting are the trusted computing environments (e.g., Trusted Platform Module (TPM), Secure Elements (SE), Trusted Execution Environments (TEEs), and Encrypted Execution Environments(E3)) discussed in the survey paper [Shepherd et al., 2016].

To guarantee any component of the Confidentiality, Integrity, and Availability (CIA) triad in CPSs, or for the purpose of detection of intelligent classes of cyber attacks, most of the existing methods require to have a secret shared between the plant and the remote controller. For example, the authors of [Noura et al., 2018] developed a physical-layer encryption algorithm for wireless machine-to-machine devices, in which sharing secret seeds is required for the implementation of the algorithm. On the other hand, the solution in [Noura et al., 2022] deals with data integrity and source authentication problems, particularly for IoT devices. The proposed message authentication algorithm requires a secret seed/key to initialize the algorithm. Similarly, it is well-understood in the CPS community that to detect intelligent coordinated networked attacks such as covert attacks [Smith, 2015], proactive detection actions must be coordinately taken in both sides of the communication channels [Ghaderi et al., 2020]. For example, moving-target [Griffioen et al., 2020] and sensor coding [Miao et al., 2016] based detection schemes implement such an idea to prevent the existence of an undetectable attack, and both require, for coordination purposes, that a secret seed is pre-shared between the plant and the controller. An anomaly detection scheme, specifically targeting differential-drive robots, is developed in [Cersullo et al., 2022], where intelligent setpoint attacks are of interest. The proposed detector leverages two command governor modules and two pseudo-random number generators (each placed on one of the two sides of the network). It has been proved that such an architecture prevents the existence of undetectable setpoint attacks only if a shared seed between the two sides of the communication channel can be established. From the above examples, it is clear that the key-establishment problem in cyber-physical systems, including mobile robots, is relevant for enhancing the security of such systems. Traditionally, the key agreement is achieved through the use of symmetric or public key cryptographic protocols [Menezes et al., 2018]. For example, using elliptic curve cryptography, in [Jain et al., 2021], the authors proposed a mutual authentication and key agreement scheme between cloud-based robots (i.e., robots that access cloud resources) and cloud servers. However, such solutions might not always be used for robotic systems. Public key protocols are computationally demanding and require a public key infrastructure [Menezes et al., 2018] and the support of a key revocation mechanism (e.g., see [Shi et al., 2021]). These requirements make public key protocol impractical for robots with limited computational capabilities [Yaacoub et al., 2021]. On the other hand, symmetric key-based solutions assume the existence of a pre-shared key. However, the compromise of such long-term keys usually leads to compromising the security of the whole system. Alternative key-establishing solutions leverage the seminal concept of wiretap channel introduced by Wyner in [Wyner, 1975]. Such schemes are not based on traditional cryptographic mechanisms. Instead, they utilize the role of noise, which is a natural characteristic in any communications system, to achieve secure communications. In particular, Wyner proved that if the communication channel between the sender and receiver is statistically better than the one from the sender to the eavesdropper, then it is possible to design an encoding mechanism to communicate with perfect secrecy. Over the years, such a concept has been leveraged to design different key-agreement protocols for CPSs, see, e.g., [Maurer, 1993, Ahlswede and Csiszár, 1993, Lara-Nino et al., 2021, Sutrala et al., 2021, Zhang et al., 2017, Rawat et al., 2017] and references therein. In [Maurer, 1993, Ahlswede and Csiszár, 1993], a key-agreement protocol based on public discussion is proposed. In [Sutrala et al., 2021], by considering a 5G-enabled industrial CPSs, a three-factor user-authenticated key agreement protocol is developed; in [Zhang et al., 2017], by using ambient wireless signals, a cross-layer key establishment model for wireless devices in CPSs is designed to allow devices to extract master keys at the physical layer. In [Rawat et al., 2017], by exploiting an information-theoretic approach, the outage probability for secrecy rate in multiple-input multiple-output (MIMO) systems for CPSs is investigated.

Remark 1 *In this thesis, for the privacy of CPSs, we refer to the secrecy/confidentiality of sensor measurements and control inputs [Darup et al., 2021].*

1.2 Thesis Goals

In a nutshell, this thesis provides solutions to the following questions:

Question 1 *How can we preserve the privacy of control inputs and sensor measurements against passive attackers if the controller is on the cloud?*

Question 2 *If homomorphic encryption is used to preserve privacy, how secure would the solution be against active attackers?*

Question 3 *How can we establish a secret key for security/privacy purposes by using a control-theoretic approach without using traditional cryptography?*

1.3 Thesis Overview and Contributions

This thesis is organized as follows:

- **Chapter 2:** All the utilized notations, required preliminary information, and definitions used alongside the thesis are defined and presented.

- **Chapter 3:** In this chapter, we propose a novel transformation-based methodology capable of preserving the privacy of the sensor measurements and control inputs when a Set-Theoretic Model Predictive Control Strategy (ST-MPC) is implemented on an honest but curious cloud. We mathematically prove that the transformed problem solved by the cloud server is equivalent to the original control problem. Also, in our proposal, the actuator is able to verify the admissibility of the received control input from the cloud server by taking advantage of the properties of ST-MPC.

- **Chapter 4:** In this chapter, we develop a novel encrypted implementation of the ST-MPC scheme, based on an additively homomorphic encryption scheme, which is capable of dealing with polyhedral state and input constraints, and bounded disturbances. Under the assumption that the cloud is honest but curious, the proposed solution guarantees the privacy of the transmitted control inputs and sensor measurements. Moreover, by using zonotopic approximations of robust one-step controllable sets (required for the ST-MPC control strategy) and by resorting to an efficient half-space projection algorithm, we can design the unavoidable plaintext computations on the actuator side to be real-time affordable for the available hardware.

- **Chapter 5:** In this chapter, we explore the use of encryption and trusted execution environments to secure plant-to-cloud communication channels and protect data and controller logic from cloud-hosted/edge-hosted CPS applications. To understand the performance implications of our approach, we also design and implement a simple prototype for a quadruple tank system [Johansson, 2000], using Intel SGX as our TEE.

- **Chapter 6:** In this chapter, we show that by exploiting the small domain of the plaintext data (e.g., sensor measurements) and the randomization process used by semantically secure encryption schemes, malware located on the plant side of the NCS is able to covertly leak sensitive information to an eavesdropper located on the measurement channel. Such information can be plaintext sensor measurements, secret encryption keys, or any other confidential data, illegitimately obtained by the malware about the operations of the control system. In simpler words, we demonstrate that the attacker is able to establish an illegitimate communication channel, also known in the literature as a covert-channel [Lampson, 1973, Abdelwahab et al., 2020]. Also, a countermeasure architecture is proposed to neutralize the introduced attacks.

- **Chapter 7:** In this chapter, we investigate the problem of sharing a secret key between a nonlinear CPS and its remote controller using a control-theoretic approach. By considering a remotely controlled wheeled mobile robot as the case of study, we extend the observer-based key-establishment solution in [Lucia and Youssef, 2022] to deal with the non-linear dynamics of mobile robots. Moreover, we experimentally validate it using a remotely maneuvered Khepera IV robot[1], the performance and the capacity of the proposed control theoretical key-agreement scheme. A demo of the performed experiment can be found at the following weblink `https://youtu.be/9FJkQhj8sdY`
- **Chapter 8:** The obtained results are summarized and future research directions are outlined.

1.4 Publications

- *A. M. Naseri, W. Lucia, M. Mannan and A. Youssef, "On Securing Cloud-Hosted Cyber-Physical Systems Using Trusted Execution Environments," 2021 IEEE International Conference on Autonomous Systems (ICAS), 2021, pp. 1-5, doi: 10.1109/ICAS49788.2021.9551155.*
- *A. M. Naseri, W. Lucia, and A. Youssef, "Confidentiality Attacks Against Encrypted Control Systems." Cyber-Physical Systems (2022): 1-20, doi: 10.1080/23335777.2022.2051209.*
- *A. M. Naseri, W. Lucia and A. Youssef, "Encrypted Cloud-Based Set-Theoretic Model Predictive Control," in IEEE Control Systems Letters, vol. 6, pp. 3032-3037, 2022, doi: 10.1109/LCSYS.2022.3182295.*
- *A. M. Naseri, W. Lucia and A. Youssef, "Encrypted Cloud-Based Set-Theoretic Model Predictive Control," in IEEE Control Systems Letters, vol. 6, pp. 3032-3037, 2022, doi: 10.1109/LCSYS.2022.3182295.*

[1] `http://www.k-team.com/khepera-iv`

Chapter 2

Notations and Background Materials

In this chapter, notations and background materials required to understand the rest of this thesis are presented. In particular, first, the utilized notation is described. Then, standard definitions and background material about set-theoretic model predictive control and homomorphic encryption are given.

2.1 Notation

We denote with $\mathbb{R}$, $\mathbb{Z}$ and $\mathbb{Z}_+$ the sets of real, integer, and non-negative integer numbers, respectively. $\mathbb{Z}_n := \{0, \ldots, n-1\}$ defines the complete residue system modulo $n \in \mathbb{Z}_+$, while $\mathbb{Z}_n^\times$ is the reduced system modulo n obtained from $\mathbb{Z}_n$ by removing all integers not relatively prime to n. The set of real-valued $n_r \times n_c$ matrices is denoted by $\mathbb{R}^{n_r \times n_c}$, while the real-valued $n_r \times 1$ column vector is denoted with $\mathbb{R}^{n_r}$. Moreover, given a matrix $M \in \mathbb{R}^{n_r \times n_c}$ and a vector $v \in \mathbb{R}^{n_r}$, M_{ij} denotes the (i, j) entry of M, while v_i denotes the $i-th$ element of v. Given a plaintext message m, $Enc[m]$ defines the corresponding ciphertext (encrypted) message according to a given encryption algorithm. Moreover, the decryption operator, namely $Dec[\cdot]$, is such that $Dec[Enc[m]] = m$. The sets of all possible plaintext (m) and ciphertexts $(Enc[m])$ messages are denoted with $\mathcal{M}$ and $\mathcal{C}$, respectively. Given two positive integer numbers $v_1, v_2 \in \mathbb{Z}_+$, then $gcd(v_1, v_2)$ and $lcm(v_1, v_2)$ and $v_1 \mod v_2$ denote the largest common positive integer divisor, the smallest positive integer common multiple and the remainder of the Euclidean division, respectively.

Given an integer $m \in \mathbb{Z}_+$, the functions $|m|$ denote the length of the binary string representing m. Given a variable v, $v(t)$ denotes the $t-$th, $t \in \mathbb{Z}_+$, sample of v obtained by sampling v with a constant sampling time $T_s > 0$. Given a set $\mathcal{S}$, $|\mathcal{S}|$ denotes the number of elements in $\mathcal{S}$. Let $r \in \mathbb{Z}_+$ be an integer number generated by a random number generator (RG), then the set of all possible values of r is denoted with $\mathcal{R}_{rg} \subset \mathbb{Z}_+$.

2.2 Preliminaries and Definitions

The following definitions, adopted from [Blanchini and Miani, 2008, Borrelli et al., 2017, Yang and Ozay, 2021], are used throughout the paper:

Definition 1 *A polyhedron $\mathcal{P} \subset \mathbb{R}^n$ is defined by the intersection of a finite number f_{hs} of half-spaces in $\mathbb{R}^n$, i.e.,*

$$\mathcal{P} := \{x \in \mathbb{R}^n \,|Hx \leq g\} \tag{1}$$

where $H \in \mathbb{R}^{f_{hs} \times n}$ and $g \in \mathbb{R}^{f_{hs}}$. A polytope is a bounded polyhedron.

Definition 2 *A zonotope $\mathcal{Z} \subset \mathbb{R}^n$ is a convex polytope which is representable as a Minkowski sum of finite line segments. Let $G = \{v_1, v_2, \ldots, v_p\}$ be a set of p generator vectors $v_j \in \mathbb{R}^n, \forall\, j$, and $c \in \mathbb{R}^n$ a center point, the generator-representation (G-rep) of $\mathcal{Z}$ is*

$$\mathcal{Z}(G, c) := \{c + \sum_{i=1}^{p} \theta_i g_i,\, \theta_i \in [-1, 1], i = 1, \ldots, p\} \tag{2}$$

Consider the following Linear-Time-Invariant (LTI) discrete-time system

$$x(k+1) = Ax(k) + Bu(k) + d(k) \tag{3}$$

where $k \in \mathbb{Z}_+ = \{0, 1, \ldots\}$, $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and the input vectors, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are the system matrices, with (A, B) controllable. Moreover, $d(k)$ is a bounded disturbance, i.e.

$$d(k) \in \mathcal{D} := \{d \in \mathbb{R}^n : H^d d \leq g^d\}, \quad 0_n \in \mathcal{D} \tag{4}$$

with $\mathcal{D}$ a compact polyhedral set, where $H^d \in \mathbb{R}^{f_d \times n}$, $g^d \in \mathbb{R}^{f_d}$. The following input and state constraints are prescribed

$$\begin{aligned} x(k) \in \mathcal{X} &:= \{x \in \mathbb{R}^n : H^x x \leq g^x\}, \quad 0_n \in \mathcal{X} \\ u(k) \in \mathcal{U} &:= \{u \in \mathbb{R}^m : H^u u \leq g^u\}, \quad 0_m \in \mathcal{U} \end{aligned} \tag{5}$$

where $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$ are compact polyhedral sets, with $H^x \in \mathbb{R}^{f_x \times n}$, $g^x \in \mathbb{R}^{f_x}$, and $H^u \in \mathbb{R}^{f_u \times n}$, $g^u \in \mathbb{R}^{f_u}$.

Definition 3 *For the system (3)-(5), the set $\mathcal{T}_0 \subset \mathcal{X}$ is said to be Robust Control Invariant (RCI) if:*

$$\forall x \in \mathcal{T}_0 \rightarrow \exists u \in \mathcal{U} : x^+ \in \mathcal{T}_0, \forall d \in \mathcal{D} \tag{6}$$

where $x^+ = Ax + Bu + d$.

Definition 4 *Consider the system (3)-(5) and a set $\mathcal{T}_i \subset \mathcal{X}$. The set of states, namely $\mathcal{T}_{i+1}$, that are robust one-step controllable (ROSC) to $\mathcal{T}_i$ is defined as*

$$\mathcal{T}_{i+1} = \{x \in \mathcal{X} : \exists u \in \mathcal{U} \; s.t. \; x^+ \in \mathcal{T}_i, \forall d \in \mathcal{D}\} \tag{7}$$

2.3 El-Gamal and Paillier Homomorphic Cryptosystems

In this section, some definitions used to describe the properties of homomorphic encryption schemes are given, and El-Gamal and Paillier cryptosystems are briefly reviewed.

Definition 5 *An encryption scheme is said homomorphic if it allows some computations on the encrypted data without access to the secret encryption key (i.e., there exists a homomorphism between the plaintext $\mathcal{M}$ and ciphertext $\mathcal{C}$ spaces) [Yi et al., 2014].*

Definition 6 *A cryptosystem is called multiplicatively homomorphic if $\forall \, m_1, m_2 \in \mathcal{M}$*

$$m_1 m_2 = Dec[Enc[m_1] \otimes Enc[m_2]] \tag{8}$$

where $\otimes$ denotes the multiplicative operator between two encrypted variables. □

Definition 7 *A cryptosystem is called additively homomorphic if $\forall\, m_1, m_2 \in \mathcal{M}$*

$$m_1 + m_2 = Dec[Enc[m_1] \oplus Enc[m_2]] \tag{9}$$

where $\oplus$ denotes the addition operation between two encrypted variables. □

2.3.1 *El-Gamal Cryptosystem*

El-Gamal is an asymmetric-key multiplicative homomorphic encryption scheme based on the difficulty of the discrete-logarithm problem [ElGamal, 1985]. The cryptosystem is characterized by the following operations:

- *Public ($\mathcal{K}_{pu}$) and private ($\mathcal{K}_{pr}$) keys generation*:

$$\mathcal{K}_{pr} = k, \quad \mathcal{K}_{pu} = \{\mathbb{G}, p, q, g, h\} \tag{10}$$

where q and $p \in \mathbb{Z}_+$ are two large randomly selected prime number satisfying $((p-1) \bmod q = 0)$, $k \in \mathbb{Z}_q$ and $g \in \mathbb{G}$, and $h = g^{\mathcal{K}_{pr}}$. $\mathbb{G} \subset \mathbb{Z}_p^\times$ is a cyclic group of the order q modulo p. - *Encryption*: A message $m \in \mathcal{M}$ is encrypted into a pair $(c_1, c_2) \in \mathcal{C}$ using $\mathcal{K}_{pu}$ and a random number $r \in \mathcal{R}_{rg} = \{1, \ldots, q-1\}$, i.e.,

$$\begin{gathered} Enc[m] = (c_1, c_2) \\ c_1 = g^r \bmod p, \quad c_2 = (m \times h^r) \bmod p \end{gathered} \tag{11}$$

- *Decryption*: An encrypted message $(c_1, c_2) = Enc[m]$ is decrypted using $\mathcal{K}_{pr}$ and $\mathcal{K}_{pu}$ as follows:

$$m = Dec[(c_1, c_2)] = (c_1^{-\mathcal{K}_{pr}} \bmod p)(c_2 \bmod p) \tag{12}$$

2.3.2 Paillier Cryptosystem

Paillier is an asymmetric-key additive homomorphic encryption scheme based on the difficulty of the integer factorization problem [Paillier, 1999]. It is characterized by the following operations:

- *Public ($\mathcal{K}_{pu}$) and private ($\mathcal{K}_{pr}$) keys generation*:

$$\mathcal{K}_{pr} = ((p-1)(q-1),\ \eta), \quad \mathcal{K}_{pu} = (n, g) \tag{13}$$

where $p \in \mathbb{Z}_+$ and $q \in \mathbb{Z}_+$ are two large and randomly selected integer prime numbers, $n = pq$ and $\eta = ((p-1)(q-1))^{-1} \bmod n^2$. A random integer number g should be selected, where $g \in \mathbb{Z}^{\times}_{n^2}$. In what follows, we assume that $g = n + 1$ [Paillier, 1999].

- *Encryption*: A message $m \in \mathcal{M}$ is encrypted into $c \in \mathcal{C}$ using $\mathcal{K}_{pu}$ and a random generated number $r \in \mathcal{R}_{rg} := \mathbb{Z}^{\times}_{n^2}$ such that $gcd(r, n) = 1$, i.e.,

$$Enc[m] = c = g^m r^n \bmod n^2 = (n+1)^m r^n \bmod n^2 \tag{14}$$

- *Decryption*: An encrypted message $c = Enc[m]$ is decrypted using $\mathcal{K}_{pr}$ as follow:

$$m = Dec[c] = \left(\frac{(c^{\mathcal{K}_{pr}} \bmod n^2) - 1}{n} \eta \right) \bmod n \tag{15}$$

Although Paillier cryptosystem is only additively homomorphic, it is also possible, exploiting the malleability of the cryptosystem, to compute multiplications between an encrypted message $Enc[m_1]$, $m_1 \in \mathcal{M}$ and a plaintext message $m_2 \in \mathcal{M}$, i.e.,

$$m_1 m_2 = Dec[Enc[m_1]^{m_2} \bmod n^2] = Dec[Enc[m_1] \odot m_2] \tag{16}$$

with $\odot$ denoting the multiplicative operator between one encrypted variable and one plaintext variable.

Remark 2 *The encryption algorithms of both El-Gamal (11) and Paillier (14) require that the random variable $r \in \mathcal{R}_{rg}$ to be freshly generated for every encryption operation by a cryptographically secure pseudorandom number generator [Ripley, 1990]. Such a requirement is necessary to ensure that these cryptosystems are semantically secure [Bellare et al., 2015]. This raises the challenge of dealing with the lack of randomness needed by a real-time CPS process. For example, in modern Unix-variants and Linux, /dev/random interface blocks until the operating system generates* *more entropy. However, such a blocking option is not acceptable in CPS applications that require real-time response. In our work, however, we focus on the case where the attacker can maliciously tamper with the RG, i.e., scenarios where the encryption protocols is vulnerable to attacks known as "random number generator attacks," see, e.g., [Goldberg and Wagner, 1996].* □

2.4 Set-Theoretic MPC (ST-MPC)

The dual-mode ST-MPC strategy developed in [Bertsekas and Rhodes, 1971, Blanchini and Miani, 2008, Angeli et al., 2008] can be summarized as follows.

Offline: Given a set of admissible initial states $x(0) \in \mathcal{X}_0 \subset \mathcal{X}$, a family of $N > 0$ ROSC sets, namely $\{\mathcal{T}_i\}_{i=1}^{N}$, is computed starting from the smallest RCI region, namely $\mathcal{T}_0$, centered in the origin. Such a family of ROSC sets can be computed as follows. First, a stabilizing control law $u(k) = -Kx(k)$ is designed for the disturbance-free plant model (3). Then, Algorithm 1 in [Rakovic et al., 2005] is used to find the smallest RCI region $\mathcal{T}_0$ associated with the previously determined control law. Finally, the terminal domain of attraction $\mathcal{T}_0$ is enlarged by means of a family of ROSC set $\{\mathcal{T}_i\}_{i=1}^{N}$ built according to the following recursive definition:

$$\begin{aligned} \mathcal{T}_i &:= \{x \in \mathcal{X} : \exists u \in \mathcal{U} \; s.t. \; x^+ \in \mathcal{T}_{i-1}, \forall d \in \mathcal{D}\} \\ &= \{x \in \mathcal{X} : \exists u \in \mathcal{U} \; s.t. \; Ax + Bu \in \tilde{\mathcal{T}}_{i-1}\}, \; i \geq 1 \end{aligned} \tag{17}$$

where $\tilde{\mathcal{T}} = \mathcal{T} \sim \mathcal{D}$, and $\sim$ denotes the Minkowski set-difference operator [Borrelli et al., 2017]. Recursion (17) is ended when the set growth saturates (i.e., $\mathcal{T}_{i-1} \not\subset \mathcal{T}_i$,) or the domain of interest is covered (i.e, $\bigcup_{i=1}^{N} \mathcal{T}_i \supseteq \mathcal{X}_0$).

Online: Given the following polyhedral representation of the offline constructed family of ROSC sets

$$\mathcal{T}_i = \{x \in \mathbb{R}^n : H_i^{\mathcal{T}} x \leq g_i^{\mathcal{T}}\}, \; H_i^{\mathcal{T}} \in \mathbb{R}^{f_i \times n}, \, g_i^{\mathcal{T}} \in \mathbb{R}^{f_i} \tag{18}$$

and

$$\tilde{\mathcal{T}}_i = \{x \in \mathbb{R}^n : H_i^{\tilde{\mathcal{T}}} x \leq g_i^{\tilde{\mathcal{T}}}\}, \; H_i^{\tilde{\mathcal{T}}} \in \mathbb{R}^{\tilde{f}_i \times n}, \, g_i^{\tilde{\mathcal{T}}} \in \mathbb{R}^{\tilde{f}_i} \tag{19}$$

the control action $u(k)$ is online computed ($\forall\, k$) as prescribed by Algorithm 1.

Algorithm 1 ST-MPC - Control Action Computation

```
Find the smallest set $i^*(k)$ containing $x(k)$ :
    $i^*(k) = \min_{\{i=1,\ldots,N\}} i \ : \ H_i^{\mathcal{T}} x(k) \leq g_i^{\mathcal{T}}$    (20)
if $i^*(k) == 0$ then
    Apply the controller associated to $\mathcal{T}_0$
else
    Compute $u(k)$ solving the optimization problem
        $u(k) = \arg\min_u \|Ax(k) + Bu\|_2^2, \quad s.t.$
        $H_{i^*(k)-1}^{\tilde{\mathcal{T}}}(Ax(k) + Bu) \leq g_{i_k^*-1}^{\tilde{\mathcal{T}}}, \ H^u u \leq g^u$    (21)
end if
```

Remark 3 *The ST-MPC strategy enjoys the following properties. The smaller RCI region $\mathcal{T}_0$ is reached at most in N steps regardless of any admissible disturbance realization, cost function, and $x(0) \in \bigcup_{i=0}^{N} \mathcal{T}_i$ [Angeli et al., 2008]. Stability and recursive feasibility are ensured regardless of the disturbance realization and cost function [Blanchini and Miani, 2008].* □

Chapter 3

A Privacy-Preserving Solution for Cloud-Enabled Set-Theoretic Model Predictive Control

This chapter proposes a solution that allows the implementation of a set-theoretic model predictive controller on the cloud while preserving its privacy. This is achieved by exploiting the offline computations of the robust one-step controllable sets used by the controller and two affine transformations of the sensor measurements and control optimization problem. It is shown that the transformed and original control problems are equivalent (i.e., the optimal control input can be recovered from the transformed one) and that privacy is preserved if the control algorithm is executed on the cloud. Moreover, we show how the actuator can take advantage of the set-theoretic nature of the controller to verify, through simple set-membership tests, if the control input received from the cloud is admissible. The correctness of the proposed solution is verified by means of a simulation experiment involving a dual-tank water system.

The solution presented in this chapter is published in the proceeding of the 2022 European Control Conference (ECC), see [Naseri et al., 2022c].

3.1 Threat Model and Problem Formulation

We consider a Networked Control System (NCS) where the controller is implemented by exploiting a third-party cloud service. We assume that the cloud has curios but honest behavior, i.e., the cloud is interested in eavesdropping on the received sensor measurements and computed control inputs, but it is not interested in affecting the performance of the NCS. Moreover, we assume that the cloud is aware of the plant model (3)-(5) and control algorithm, but it has no prior knowledge about the initial state of the system or the desired equilibrium point.

Assumption 1 *The initial state of the system and the desired equilibrium point are assumed to be secret.*

The objective of this work can be stated as follows. Design a networked control system architecture where the ST-MPC controller can be implemented on the cloud while achieving the following goals:

- *Data confidentiality:* the state vector and control inputs are kept secret from the cloud.
- *Efficient computations:* the computation demand on the plant side is modest.
- *Minimize the communication overhead:* the introduced communication overhead is minimal or null w.r.t. standard NCSs.
- *Verification:* the admissibility of the received control input can be easily verified on the plant's side.

Note that the last goal is not strictly necessary for the considered passive thread model. Nevertheless, it is desirable in any networked CPSs to be able to detect the possibility that active attackers (either on the cloud on in the communication channels) attempt to corrupt the control system operations [Dibaji et al., 2019].

3.2 Proposed Solution

To achieve the desired level of confidentiality, we propose implementing on the cloud a transformed version of ST-MPC. To this end, two affine and random transformations [Shan et al., 2018]

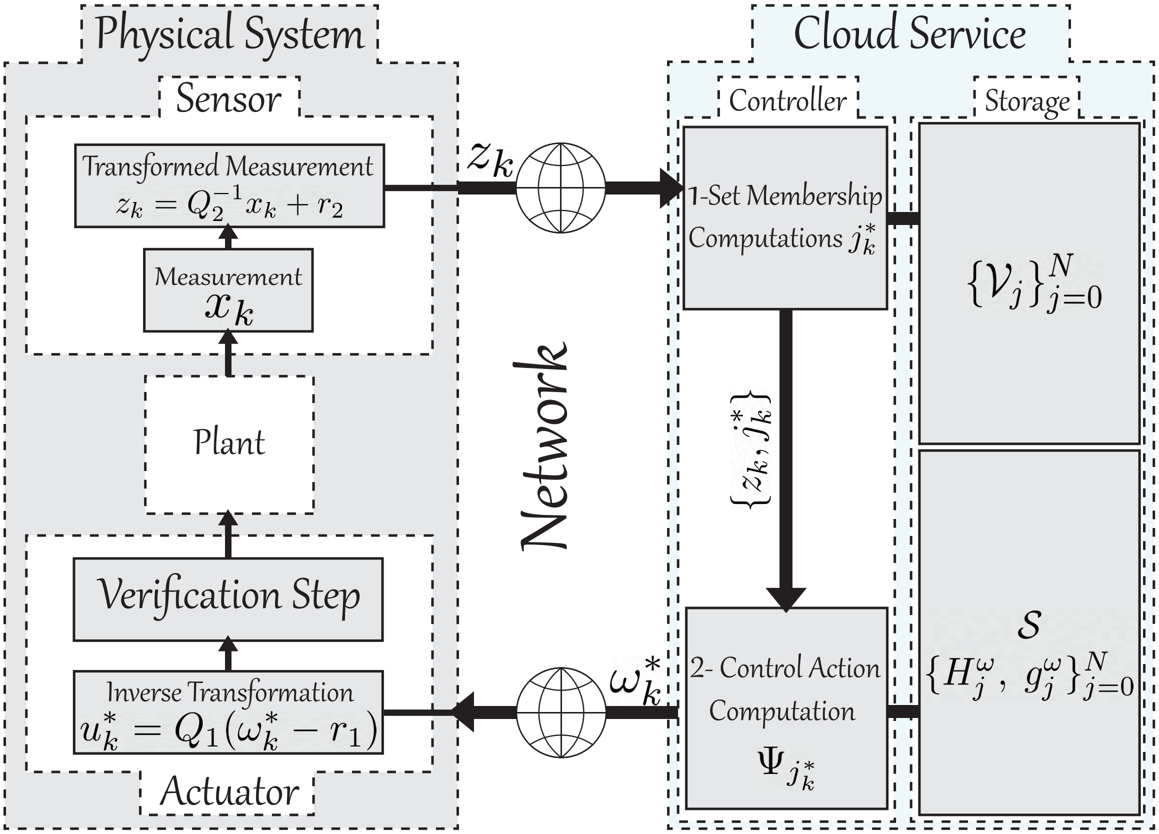


Figure 3.1: Proposed NCS architecture for the private computation of the ST-MPC on the Cloud.

of u_k and x_k, are considered:

$$\omega_k = Q_1^{-1} u_k + r_1 \tag{22a}$$

$$z_k = Q_2^{-1} x_k + r_2 \tag{22b}$$

where $Q_1 \in \mathbb{R}^{m \times m}$, $Q_2 \in \mathbb{R}^{n \times n}$ are random non-singular matrices, and $r_1 \in \mathbb{R}^m$, $r_2 \in \mathbb{R}^n$ random vectors.

Remark 4 *In [Xu and Zhu, 2015], it has been shown that the control input transformation (22a) might be sufficient to obtain a private implementation on the cloud of a standard MPC optimization problem. In the proposed solution, two transformations are instead proposed for the following reasons:*

- *Minimize the communication and computation overhead on the sensor's side of the NCS.*

- *Compute on the cloud the set-membership index i_k^* in* (20). □

In what follows, the proposed architecture (see Fig. 3.1) is designed assuming that z_k is sent by the sensor to the networked controller and ω_k from the cloud to the actuator. Therefore, the transformation (22b) is applied to x_k on the sensor's side of the NCS, and the inverse transformation of (22a) is used by the actuator to recover u_k from ω_k.

3.2.1 Set-membership computation on the cloud

Since only the transformed state measurement vector z_k is available on the cloud, then the set-membership identification (20) can be performed only if the family of robust one-step controllable sets $\{\mathcal{T}_i\}_{i=0}^N$ has been offline transformed according to (22b) and stored in the cloud.

Proposition 1 *Consider the family of robust one-step controllable sets $\{\mathcal{T}_i\}_{i=0}^N$ computed as in* (17)*. For any $x_k \in \bigcup_{j=1}^N$, then i_k^* (computed solving* (20)*) is equal to the optimal solution j_k^* of the following transformed optimization problem*

$$j_k^* = \min_{j=1,\ldots,N} \eta \quad s.t., \;\; \Gamma_j^z z_k \leq \gamma_j^z \tag{23}$$

where the constraint $\Gamma_j^z z_k \leq \gamma_j^z$ denotes a polyhedral set $\mathcal{V}_j$ obtained applying to $\mathcal{T}_j$ the affine transformation (22b)*, i.e.,*

$$\mathcal{T}_j \xrightarrow{(Q_2,r_2)} \mathcal{V}_{\mathcal{J}}= \{z \in \mathbb{R}^n : H_j^x Q_2 z \leq g_j^x + H_j^x Q_2 r_2\} = \{z \in \mathbb{R}^n : \Gamma_j^z z \leq \gamma_j^z\} \tag{24}$$

where $\Gamma_j^z = H_j^x Q_2$ and $\gamma_j^z = g_j^x + H_j^x Q_2 r_2$.

Proof 1 *The proposition can be demonstrated by resorting to a proof by contradiction (Reductio ad Absurdum). Assume that $i_k^* < j_k^*$. By resorting to simple manipulations and by recalling that $z_k = Q_2^{-1} x_k + r_2$, we can write the following equivalent relations*

$$\begin{aligned} x_k \in \mathcal{T}_{i_k^*} &\Rightarrow H_{i_k^*}^x x_k \leq g_{i_k^*}^x \Rightarrow H_{i_k^*}^x Q_2 Q_2^{-1} x_k \leq g_{i_k^*}^x, \\ &\Rightarrow H_{i_k^*}^x Q_2 Q_2^{-1} x_k + H_{i_k^*}^x Q_2 r_2 \leq g_{i_k^*}^x + H_{i_k^*}^x Q_2 r_2 \Rightarrow H_{i_k^*}^x Q_2 z_k \leq g_{i_k^*}^x + H_{i_k^*}^x Q_2 r_2 \\ &\Rightarrow \Gamma_{i_k^*}^z z_k \leq \gamma_{i_k^*}^z \end{aligned} \tag{25}$$

where the conclusive implication $x_k \in \mathcal{T}_{i_k^*} \Rightarrow \Gamma^z_{i_k^*} z_k \leq \gamma^z_{i_k^*}$ represents an absurd stating that $z_k \in \mathcal{V}_{i_k^*}$ and that j_k^* is not the optimal solution of (23)*. Along the same lines, the assumption that $i_k^* > j_k^*$ defines another absurd where i_k^* is not the optimal solution of* (20)*. Therefore, we can conclude that $i_k^* = j_k^*$.*

3.2.2 Control action computation on the cloud

To preserve the privacy of the optimization (21), the control problem must be properly transformed such that the input state vector is z_k (instead of x_k) and the output is ω_k (instead of u_k). To this end, it is necessary to apply the affine transformations (22a)-(22b) to the family of disturbance-free one-step controllable sets $\{\tilde{\mathcal{T}}_j\}_{i=0}^N$ and input constraint set $\mathcal{U}$:

$$\tilde{\mathcal{T}}_j \xrightarrow{(Q_2,r_2)} \tilde{\mathcal{V}}_j = \{z \in \mathbb{R}^n : \tilde{H}^x_j Q_2 z \leq \tilde{g}^x_j + \tilde{H}^x_j Q_2 r_2\} \tag{26}$$

$$\begin{aligned} \mathcal{U} \xrightarrow{(Q_1,r_1)} \Omega &= \{\omega \in \mathbb{R}^m : H^u Q_1 \omega \leq g^u + H^u Q_1 r_1\} \\ &= \{\omega \in \mathbb{R}^m : H^\omega \omega \leq g^\omega\} \end{aligned} \tag{27}$$

where $H^\omega = H^u Q_1$ and $g^\omega = g^u + H^u Q_1 r_1$.

Proposition 2 *Under the affine transformations* (22a)-(22b)*, the optimization problem Φ_j in* (21) *is recast into an optimization problem Ψ_j defined on z_k and ω_k:*

$$\Phi_j \xrightarrow[(Q_2,r_2)]{(Q_1,r_1)} \Psi_j = \begin{cases} \omega_k^* = \arg\min\limits_{\omega} J_j(z_k, \omega) \ s.t. \\ \tilde{H}^z_{j-1} \omega \leq \tilde{g}^z_{j-1}, \ H^\omega \omega \leq g^\omega \end{cases} \tag{28}$$

where

$$\begin{gathered} J_j(z_k, \omega) = \|AQ_2 z_k + BQ_1 \omega_k - AQ_2 r_2 - BQ_1 r_1\|_2^2, \\ \tilde{H}^z_{j-1} = \tilde{H}^x_{j-1} BQ_1, \\ \tilde{g}^z_{j-1} = \tilde{g}^x_{j-1} - \tilde{H}^x_{j-1} AQ_2 (z_k + r_2) + \tilde{H}^x_{j-1} BQ_1 r_1 \end{gathered} \tag{29}$$

Proof 2 *The cost function $J_j(z_k, \omega)$ can be simply obtained transformed the cost function $\|Ax_k + Bu\|_2^2$ according to* (22a)-(22b)*. Moreover, $x_{k+1} \in \tilde{\mathcal{T}}_{j-1}$ implies that $\tilde{H}^x_{j-1} Q_2 z_{k+1} \leq \tilde{g}^x_{j-1} + \tilde{H}^x_{j-1} Q_2 r_2$ (see* (26)*). Then, by noting that $z_{k+1} = Q_2^{-1} AQ_2 z_k + Q_2^{-1} BQ_1 \omega_k - Q_2^{-1} AQ_2 r_2 -$*

$Q_2^{-1}BQ_1r_1 + r_2$, we can re-write the constraint $\tilde{H}^x_{j-1}Q_2z_{k+1} \leq \tilde{g}^x_{j-1} + \tilde{H}^x_{j-1}Q_2r_2$ as

$$\tilde{H}^x_{j-1}BQ_1\omega_k \leq \tilde{g}^x_{j-1} - \tilde{H}^x_{j-1}AQ_2(z_k + r_2) + \tilde{H}^x_{j-1}BQ_1r_1 \tag{30}$$

As a consequence, the transformation

$$\tilde{H}^x_{j-1}(Ax_k + Bu) \leq \tilde{g}^x_{j-1} \xrightarrow[(Q_2,r_2)]{(Q_1,r_1)} \tilde{H}^z_{j-1}\omega \leq \tilde{g}^z_{j-1}$$

holds true. Finally, given (27), the transformation

$$H^u u \leq g^u \xrightarrow[(Q_2,r_2)]{(Q_1,r_1)} H^\omega \omega \leq g^\omega$$

concludes the proof.

Proposition 3 *Let ω^*_k be the optimal solution of the optimization problem Ψ_i. Then, $u_k = Q_1(\omega^*_k - r_1)$ is the optimal solution of the optimization problem Φ_i.*

Proof 3 *First, $u_k = Q_1(\omega^*_k - r_1)$ is an admissible solution for Φ_i. Indeed, the set of admissible solutions of Φ_i is given by*

$$\tilde{H}^x_{j-1}(Ax_k + Bu) \leq \tilde{g}^x_{j-1}, \;\; H^u u \leq g^u \tag{31}$$

By replacing $u_k = Q_1(\omega^*_k - r_1)$, and by performing simple manipulations, both constraints in (31) can be rewritten as follows:

(i) $\tilde{H}^x_{j-1}(Ax_k + Bu_k) \le \tilde{g}^x_{j-1}$*, is equivalent to*

$$\begin{aligned}
&\tilde{H}^x_{j-1}BQ_1(\omega^*_k - r_1) \le \tilde{g}^x_{j-1} - \tilde{H}^x_{j-1}Ax_k \\
&\Rightarrow \tilde{H}^x_{j-1}BQ_1\omega^*_k \le \tilde{g}^x_{j-1} - \tilde{H}^x_{j-1}Q_2Q_2^{-1}Ax_k - \tilde{H}^x_{j-1}AQ_2r_2 + \tilde{H}^x_{j-1}AQ_2r_2 + \tilde{H}^x_{j-1}BQ_1r_1 \\
&\Rightarrow \tilde{H}^x_{j-1}BQ_1\omega^*_k \le \tilde{g}^x_{j-1} - \tilde{H}^x_{j-1}AQ_2(Q_2^{-1}x_k + r_2) + \tilde{H}^x_{j-1}AQ_2r_2 + \tilde{H}^x_{j-1}BQ_1r_1 \\
&\Rightarrow \tilde{H}^x_{j-1}BQ_1\omega^*_k \le \tilde{g}^x_{j-1} - \tilde{H}^x_{j-1}AQ_2z_k + \tilde{H}^x_{j-1}AQ_2r_2 + \tilde{H}^x_{j-1}BQ_1r_1 \\
&\Rightarrow \tilde{H}^x_{j-1}BQ_1\omega_k \le \tilde{g}^x_{j-1} - \tilde{H}^x_{j-1}AQ_2(z_k + r_2) + \tilde{H}^x_{j-1}BQ_1r_1 \\
&\Rightarrow \tilde{H}^z_{j-1}\omega^*_k \le \tilde{g}^z_{j-1}
\end{aligned} \tag{32}$$

(ii) $H^u u \le g^u$ *is equivalent to*

$$\begin{aligned}
& H^uQ_1(\omega^*_k - r_1) \le g^u \\
\Rightarrow\ & H^uQ_1\omega^*_k \le g^u - H^uQ_1r_1 \\
\Rightarrow\ & H^\omega\omega^*_k \le g^\omega
\end{aligned} \tag{33}$$

As a consequence, (31) *is equivalent to*

$$\tilde{H}^z_{j-1}\omega^*_k \le \tilde{g}^z_{j-1},\ H^\omega\omega^*_k \le g^\omega \tag{34}$$

which is by hypothesis satisfied.

Then, we can resort to a proof by contradiction to show that $u_k = Q_2(\omega^*_k - r_2)$ *is the optimal solution of* (21)*. Assume that* $\mu_k \ne Q_2(\omega^*_k - r_2)$ *is the optimal solution of* (21)*. Then, for the optimality condition, we have that*

$$\|Ax_k + B\mu_k\|_2^2 \le \|Ax_k + BQ_2(\omega^*_k - r_2)\|_2^2$$

that resorting to similar manipulations to the ones used in (32) *can be re-written as*

$$\|AQ_2z_k + B_1Q_1(Q_1^{-1}\mu_k + r_1) - AQ_2r_2 - BQ_1r_1\|_2^2 \le \|Az_k + B\omega^*_k - AQ_2r_2 - BQ_1r_1\|_2^2$$

which defines an absurd in which the optimization (28) *might obtain a better solution for* $\omega_k =$

$Q_1^{-1}\mu_k + r_1 \neq \omega_k^*$.

3.2.3 Information stored in the cloud

To compute in the cloud the control action ω_k^* the optimization problems (23) and (28) must be implemented. One possibility is to entirely transmit, at each k, the two optimization problems, see, e.g., [Xu and Zhu, 2015]. However, such a solution suffers from unavoidable and undesirable communication and computation overhead on the measurement channel and the plant's side of the NCS. On the other hand, a more practical solution consists of offline uploading on the cloud part of the optimization problem.

In particular, to implement (23), the family of transformed controllable sets $\{\mathcal{V}_i\}_{i=1}^N$ can be offline stored in the cloud. On the other hand, to implement (28), the following set of information must be uploaded

$$\mathcal{S} := \begin{cases} \{AQ_2, BQ_1, -AQ_2r_2 - BQ_1r_1, \tilde{H}_i^x AQ_2\} \\ \text{and} \\ \{\tilde{H}_i^z, \tilde{g}_i^x - \tilde{H}_i^x AQ_2r_2 + \tilde{H}_i^x BQ_1r_1\}_{i=0}^N \end{cases}$$

Remark 5 *Please note that the optimization problem* (28) *cannot be completely pre-loaded in the cloud since $\tilde{g}_i^z$ is a function of z_k and, as a consequence, of x_k. Nevertheless, given the stored information set $\mathcal{S}$ and z_k (sent by the sensor), the cloud can at each time instant k construct the optimization* (28) *and solve it.*

3.2.4 Control input verification

The actuator receives the transformed control input ω_k^*. Therefore, the actuator is in charge of applying the transformation $u_k^* = Q_1(\omega_k^* - r_1)$ and applying u_k^* to the plant. Besides, by exploiting the fact that the robust one-step controllable sets are by construction nested [Blanchini and Miani, 2008], i.e. $\mathcal{T}_i \subset \mathcal{T}_{i+1}, \forall\, i = 0, \ldots, N-1$, the actuator can perform simple set-membership checks to verify the admissibility of u_k. In particular, by taking a worst-case approach, the set-membership index i_0^* can be assumed equal to N. Then, at each time instant, i_k^* is upper bounded by $\bar{i}_k = max(1, N-k)$. As a consequence, the actuator can verify the admissibility of u_k checking if $u_k \in \mathcal{U}$ and $Ax_k + Bu_k^* \in \mathcal{T}_{\bar{i}_k-1}$.

3.3 Performance and Security Analysis

3.3.1 Security analysis

In the proposed control architecture, the data exchange between the cloud and the plant includes the variables $z_k = Q_2^{-1}x_k + r_2$ (on the measurement channel) and $\omega_k = Q_1^{-1}u_k + r_1$ (on the actuation channel). Since the matrices Q_1, Q_2 and the vectors r_1, r_2 are randomly selected and unknown to the adversary, then z_k and ω_k do not provide meaningful information about the real state evolution of the system and control inputs. One possibility for the attacker to recover Q_1, r_1, Q_2, r_2 is to leverage proper prior information about the expected evolution of the system to lunch a so-called known-plaintext attack.Under the considered assumptions (see Section 3.1), the attacker is not aware of the initial state condition and desired equilibrium point. As a consequence, such an attack is not doable. Moreover, the attack proposed in [Laud and Pankova, 2013], against the transformation-based linear programming outsourcing problem, does not apply to our control architecture. Two main reasons can be stated. First, the attack, to be successful, requires the presence of specific constraints in the form $u_k \geq 0$ acting on the decision variable. Second, it requires the shift vector r_1 to be a positive (component-wise) vector in order to verify the admissibility of candidate u_k values (such a limitation is not imposed in (22a)-(22b)).

3.3.2 Computational Overhead

The sensor is only in charge of computing the transformation (22b) starting from the state measurement vector x_k. The complexity of this operation is $O(n^2)$ with n the number of system states. On the other hand, the actuator is in charge of applying the inverse transformation of (22a) to the received input ω_k^*. This operation has the complexity of $O(m^2)$ where m is the size of the control input vector. Furthermore, the actuator is responsible for the verification step. The computational complexity of this verification step is $O(p_i)$, where p_i is the number of inequalities representing $\mathcal{T}_i$.

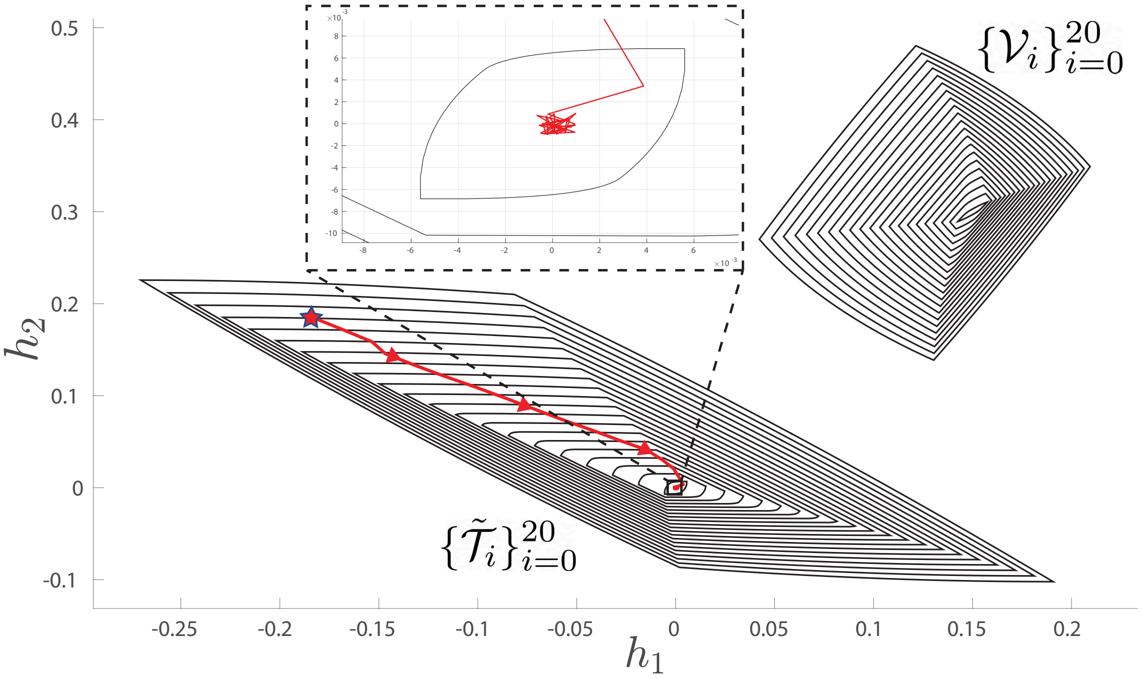


Figure 3.2: State trajectory and robust one-step controllable sets.

3.3.3 Communication Overhead

The proposed control architecture (Fig. 3.1) does not introduce any additional communication overhead w.r.t. standard NCSs. This finds justifications in the fact that the transmitted variables z_k and ω_k^* have the same size of x_k and u_k^*.

3.4 Simulation Results

To verify the correctness and functionality of our proposed model, we implement the ST-MPC controller with our proposed privacy-preserving mechanism on the Two-Tank water system testbed model used in [Bemporad et al., 1997]. The states of the system are the level of water inside the tanks i.e. $x = [h_1, h_2]^T$, while the control input vector is $u = [u_p, u_l, u_u]^T$, consisting of the input voltage of the pump u_p, lower interconnect valve u_l and upper interconnect valve u_u. The nonlinear continuous-time dynamics have been linearized around the equilibrium pair $x_{eq} = [0.5, 0.5]^T$ and $u_{eq} = [0.938, 1, 0.833]^T$ and discretized with a sampling time of $T_s = 1\ sec$. The fully observable linearized model (3) has the following system matrices.

$$A = \begin{bmatrix} 0.993 & 0.003 \\ 0.007 & 0.982 \end{bmatrix}, B = \begin{bmatrix} 0.008 & -0.003 & -0.003 \\ 0.000 & 0.003 & 0.003 \end{bmatrix}$$

The process disturbance set is $\mathcal{D} = \{d \in \mathbb{R}^2 : |d_j| \leq 0.001,\ j = 1, 2\}$, while the state and input constraints are $-0.5 \leq u_p \leq 1.5$, $-0.25 \leq u_l \leq 1.75$, $-0.8 \leq u_u \leq 1.2$ and $0.02 \leq h_1,\ h_2 \leq 0.8$. To implement the ST-MPC controller we have built $N = 20$ robust one-step controllable sets $\mathcal{T}_i$. The optimization transformation pairs $\{Q_1, r_1\}$ and $\{Q_2, r_2\}$ have been randomly chosen ensuring that Q_1, Q_2 are invertible matrices:

$$\begin{gathered} Q_1 = \begin{bmatrix} 0.647 & 0.042 & 0.477 \\ 0.075 & 0.036 & 0.437 \\ 0.133 & 0.836 & 0.936 \end{bmatrix}, \quad r_1 = \begin{bmatrix} 2 \\ -1 \\ 5 \end{bmatrix} \\ Q_2 = \begin{bmatrix} 1.123 & -1.319 \\ -2.201 & 0.901 \end{bmatrix}, \quad r_2 = \begin{bmatrix} 0.15 \\ 0.3 \end{bmatrix} \end{gathered} \tag{35}$$

The simulation results have been obtained considering an initial state $x_0 = [-0.2, 0.2]^T$ and a time frame of 50 seconds. Fig. 3.2 shows the system's state trajectory, the family of robust controllable sets $\{\mathcal{T}_i\}_{i=0}^N$ and its affine transformation $\{\mathcal{V}_i\}_{i=0}^N$ (offline uploaded on the cloud for set-membership computation purposes). Moreover, it is possible to appreciate that given $\{\mathcal{V}_i\}_{i=0}^N$ and z_k, the adversary is not able to understand the state evolution of the system. Fig. 3.4 shows the set-membership index computed by the cloud (i.e., j_k^*). This index starts from $j_0^* = 19$ and presents a monotonically decreasing behavior which testifies that the control algorithm is properly working according to the prescriptions of the standard ST-MPC scheme [Angeli et al., 2008]. Finally, Fig. 3.3 shows that the cloud-enabled ST-MPC preserves the prescribed input constraints.

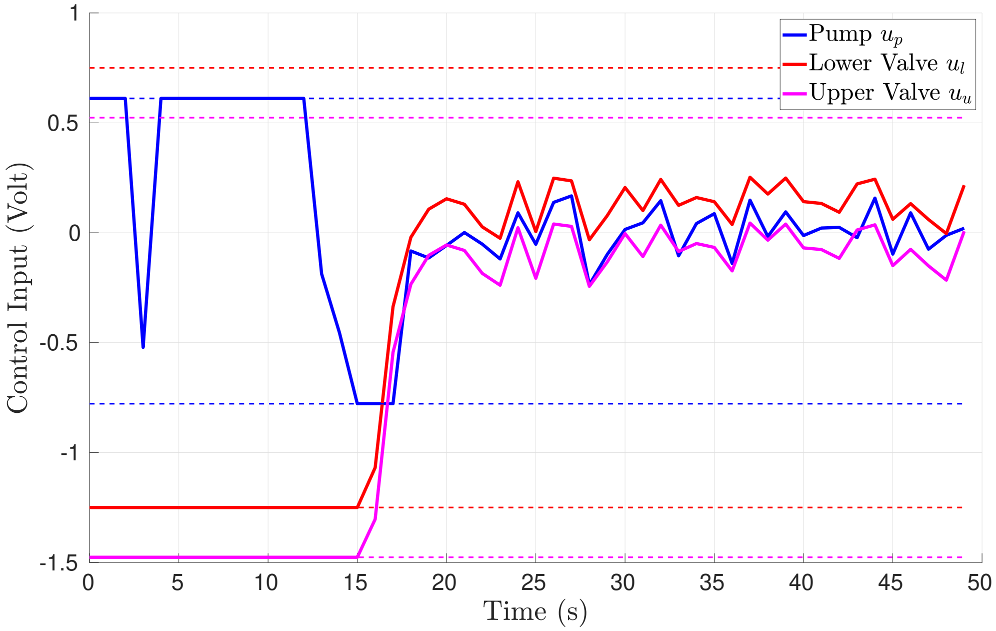


Figure 3.3: Control inputs.

3.5 Conclusions

We proposed a networked control architecture for cloud-enabled ST-MPC controllers, where the confidentiality of the sensor measurements and control inputs is preserved. This has been achieved by means of a proper random transformation of the operations performed by the controller. In comparison to [Alexandru et al., 2018, Darup et al., 2018, Schlüter and Darup, 2020], the proposed solution imposes less computational overhead on the sensors and actuators. Differently from [Xu and Zhu, 2015], the proposed approach is able to deal with plants subject to both input and state constraints, and it does not require the transmission, at each time step, to the cloud of the transformed control optimization problem. Furthermore, while in [Xu and Zhu, 2015] the computational complexity of the control input verification step (at the actuator's side of the NCS) increases with the prediction horizon, in the proposed solution, such a test can be simply carried on resorting to simple set-membership tests involving the robust one-step evolution of the system. Furthermore, the proposed strategy does not require extra computations on the cloud to enable the control input verification on the actuator's side. In the future it would be interesting to extend/enhance this scheme to be secure for situations where the adversary might be able to launch known/chosen plaintext attacks.

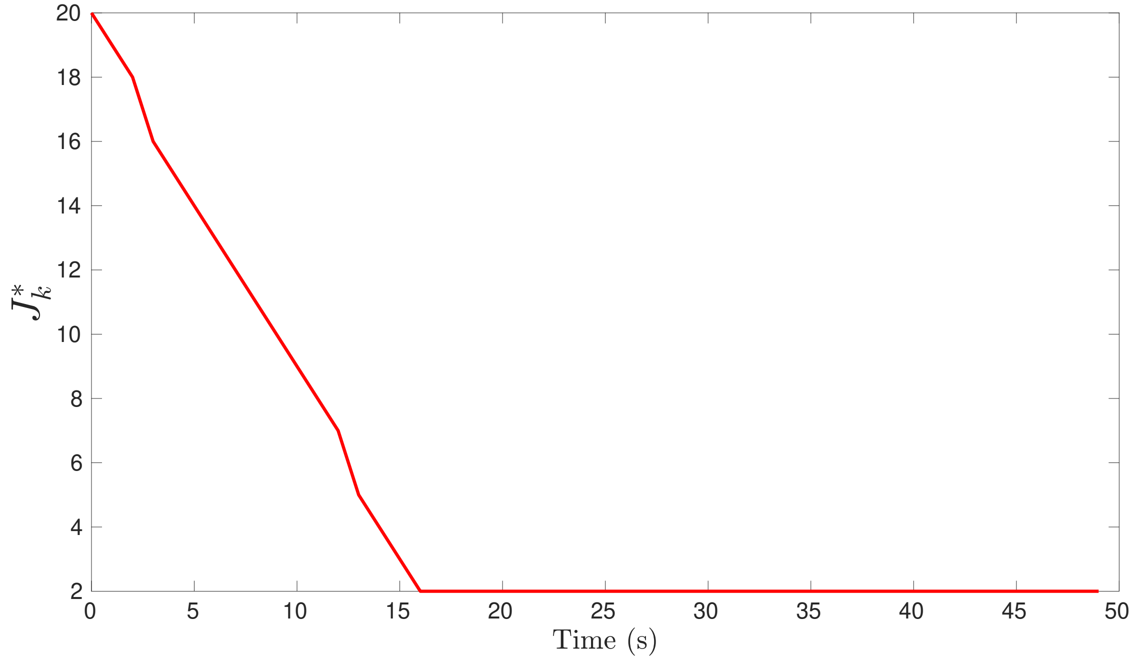


Figure 3.4: Set-membership index j_k^*.

Chapter 4

Encrypted Cloud-Based Set-Theoretic Model Predictive Control

In this chapter, we propose an encrypted set-theoretic model predictive control (ST-MPC) strategy for cloud-based networked control systems. In particular, we consider a scenario where the plant is subject to state and input constraints, and a curious but honest cloud provider is available to implement the control logic remotely. We address the inherent privacy issue by jointly using an additive homomorphic cryptosystem and a modified version of the ST-MPC algorithm, which is tailored to run on encrypted data. We show that, by leveraging a family of zonotopic inner approximations of robust one-step controllable sets and a half-space projection algorithm, we can design the unavoidable computational load on the smart actuator's side to be real-time affordable by the available hardware compared to other existing solutions. A simulation experiment, considering a two-tank water system, is presented to verify the effectiveness of the proposed approach.
The solution proposed in this chapter is published in the IEEE Control Systems Letters (L-CSS) journals and accepted for presentation at the 61st IEEE Conference on Decision and Control (CDC) in December of 2022, see [Naseri et al., 2022b].

4.1 Background and Problem Formulation

In this section, first, background material on Proximal Gradient Methods (PGM) is reviewed. Then, the problem of interest is stated.

4.1.1 Proximal Gradient Method (PGM)

PGM is a technique to solve optimization problems in the following form [Parikh and Boyd, 2014]:

$$\min_{u} h(u,x), \quad h(u,x) := f(u,x) + g(u,x) \tag{36}$$

where $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are closed proper convex functions and f is differentiable [Parikh and Boyd, 2014]. By defining the PGM operator as follows

$$\text{prox}_{\alpha,g}(v,x) := \arg\min_{u} g(u,x) + \frac{1}{2\alpha} \parallel u - v \parallel_2^2 \tag{37}$$

where $\alpha > 0$ denotes the step size, then, the solution of (36) can be found by successive iterations of the following PGM

$$\begin{array}{rcl} u^{j^+} & = & u^j - \alpha \nabla_u f(u^j, x) \\ u^{j+1} & = & \text{prox}_{\alpha,g}(u^{j^+}, x) \end{array} \quad , \quad j \geq 0 \tag{38}$$

where u^0 denotes the initial guess.

For an appropriate choice of α (e.g., $\alpha \in (0, 1/L]$ if ∇f is Lipschitz continuous with constant L [Sohrab, 2003, Theorem 4.6.3]), the above iterations are guaranteed to converge to the optimal solution in a finite number of steps.

4.1.2 Problem Formulation

Consider a Networked Control System (NCS) where a HE system fulfilling (8)-(9) is used to secure the communication channels between the plant and the cloud provider (see Fig. 4.1). We assume that the control logic is implemented on an honest but curious cloud, i.e., the cloud provider does not deviate from the expected protocol but attempts to learn from the received messages. Such privacy leakage could form the foundation of more complex attacks. Moreover, we assume that the

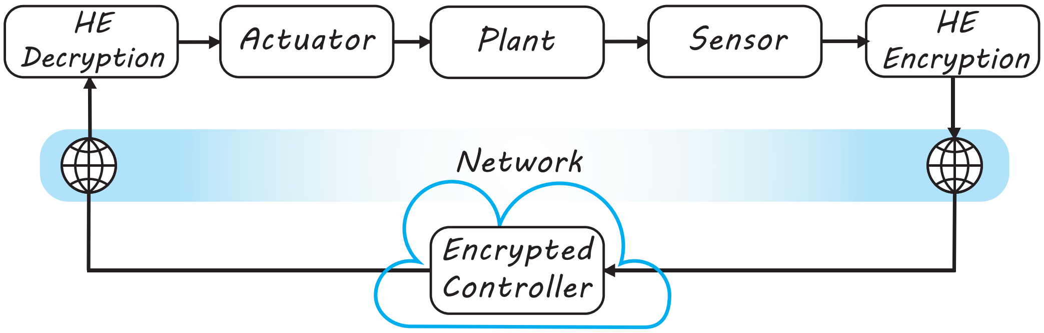


Figure 4.1: Cloud-based Networked Control System with HE.

actuator is capable of performing simple arithmetic operations.

The objectives of this chapter can be stated as follow:

Design an encrypted version of the ST-MPC controller, namely E-ST-MPC, such that the following objectives are satisfied:

- *(O1) The E-ST-MPC can be implemented on the cloud preserving the privacy of the state measurement vector $x(k)$ and control input $u(k)$.*
- *(O2) The E-ST-MPC strategy enjoys the same properties of ST-MPC (see Remark 3).*
- *(O3) The number of arithmetic operations required on the actuator can be designed to be real-time affordable by the available hardware.*

4.2 Proposed Solution

The proposed solution takes advantage of the available HE cryptosystem to implement the operations of ST-MPC on the encrypted measurement state vector and the cloud. To this end, by referring to ST-MPC operations in Algorithm 1, the following issues must be properly addressed:

- The set membership evaluation (20) involves inequality checks that cannot be evaluated on the encrypted data.
- As shown in different papers, see, e.g., [Darup et al., 2019], if only one cloud is available, it is not possible to obtain the optimal solution of (21) on the encrypted data.

- The number of half-spaces used to represent the ROSC sets (17) cannot be controlled and it increases with the number of computed sets [Blanchini and Miani, 2008].

In what follows, first, we exploit set-theoretic arguments to define a computable worst-case upper bound on the realization of set-membership signal $i^*(k)$. Then, we jointly use one-iteration of the PGM and a halfspace projection to compute admissible, although not optimal, control actions that preserve the ST-MPC properties (see Remark 3). Finally, we develop an ad-hoc inner zonotopic approximation of the ROSC sets to upper-bound and control the maximum number of operations required on the actuator.

4.2.1 E-ST-MPC

Following [Darup et al., 2017], a simple way to compute the set membership index (20) would prescribe its evaluation on the sensor (before encryption) and transmission (unencrypted) to the cloud. Although effective, this solution requires further computations on the sensors' side, which is typically not possible. Moreover, such an approach discloses the set-membership information to eavesdroppers on the measurement channel and the cloud. Therefore, we propose a different privacy-preserving solution leveraging the fact that the computed family of ROSC sets $\{\mathcal{T}_i\}$ is, by construction, nested, i.e., $\mathcal{T}_i \subset \mathcal{T}_{i+1}, \forall\, i$ [Blanchini and Miani, 2008]. In particular, such a property allows for upper bound $i^*(k)$ on the cloud with the following monotonically decreasing function

$$\bar{i}(k) = \max(\bar{i}(k-1) - 1, 0), \quad \bar{i}(0) = N \tag{39}$$

The optimal solution of (21) can be obtained using different optimization strategies. Of interest here is the PGM method [Parikh and Boyd, 2014] considered in [Darup et al., 2018]. In particular, the optimization (21) is equivalent to (36) if

$$\begin{aligned} f(u,x) &= \|Ax(k) + Bu(k)\|_2^2 \\ g(u,x) &= \mathcal{I}_{\mathcal{U}_i(x(k))}(u) := \begin{cases} 0 & \text{if } u \in \mathcal{U}_i(x(k)) \\ \infty & \text{Otherwise} \end{cases} \end{aligned} \tag{40}$$

where $\mathcal{I}_{\mathcal{U}_i(x(k))}$ is known as the indicator function of the set $\mathcal{U}_i(x(k))$, with

$$\mathcal{U}_i(x(k)) = \left\{ u \in \mathbb{R}^m : H^{\mathcal{U}(x(k))} u \leq g^{\mathcal{U}(x(k))} \right\} \tag{41}$$

where

$$H_i^{\mathcal{U}(x(k))} = \begin{bmatrix} H^{\tilde{\mathcal{T}}}_{i^*(k)-1} B \\ H^u \end{bmatrix} \in \mathbb{R}^{(\tilde{f}_i + f_u) \times m} \tag{42}$$

$$g_i^{\mathcal{U}(x(k))} = \begin{bmatrix} g^{\tilde{\mathcal{T}}}_{i^*(k)-1} - H^{\tilde{\mathcal{T}}}_{i^*(k)-1} A x(k) \\ g^u \end{bmatrix} \in \mathbb{R}^{(\tilde{f}_i + f_u)} \tag{43}$$

Moreover, since $g(u, x)$ is an indicator function, then the PGM operator (37) is equivalent to the following projection

$$\text{Proj}_{\mathcal{U}_i(x(k))}(u^{j^+}) := \underset{u \in \mathcal{U}_i(x(k))}{\arg\min} \; \|u - u^{j^+}\|_2^2 \tag{44}$$

with $u^{j^+} = u^j - \alpha \nabla_u f(u^j, x(k))$.

However, as discussed in, e.g., [Darup et al., 2018], the PGM algorithm cannot be entirely executed on the encrypted data using a single cloud. Typically, only a single iteration of (38) can be performed [Parys and Pipeleers, 2018], i.e. $u^0 \to u^{0^+} \to u^1$, with $u(k) = u^1 \in \mathcal{U}$ as the sub-optimal control input applied to the plant. In particular,

$$\begin{aligned} u^{0^+} &= u^0 - \alpha \nabla_u f(u^0, x(k)) \\ &= u^0 - \alpha((2B^T B)u^0 + (2B^T A)x(k)) \\ &= (I - \alpha(2B^T B))u^0 - \alpha(2B^T A)x(k) \end{aligned} \tag{45}$$

can be computed on the cloud on the encrypted state measurement as follows:

$$\begin{aligned} Enc[u^{0^+}] = \; & ((I - \alpha(2B^T B)) \odot Enc[u^0]) \\ & \oplus((-\alpha 2B^T A) \odot Enc[x(k)]) \end{aligned} \tag{46}$$

where $\alpha \in (0, 2/\lambda_{max}(2B^T B))$. On the other hand,

$$u^1 = \text{Proj}_{\mathcal{U}_i(x(k))}(u^{0^+}) := \underset{u \in \mathcal{U}_i(x(k))}{\arg\min} \|u - u^{0^+}\|_2^2 \tag{47}$$

can be computed on the actuator after decryption. To numerically solve (47), we propose using the halfspace projection method defined in [Bauschke, 1996]. By denoting with $h(j)$ and $g(j)$ the $j - th$ row of $H_i^{\mathcal{U}(x(k))}$ and $g_i^{\mathcal{U}(x(k))}$, respectively, the projection (47) is performed using the following recursion

$$\begin{array}{rcl} v^0 & = & u^{0^+} \\ v^{j+1} & = & v^j - \frac{\max(h(j)v^j - g(j), 0)}{\|h(j)\|_2} h(j)^T, \ 0 \leq j \leq \tilde{f}_i + f_u \\ u^1 & = & v^{\tilde{f}_i + f_u + 1} \end{array} \tag{48}$$

According to the above procedure, the time complexity of computations required on the actuator is $\mathcal{O}(\tilde{f}_i + f_u)$, where $\tilde{f}_i + f_u$ is the number of inequalities used to describe the polyhedral set $\mathcal{U}_i(x(k))$. Note that while f_u is constant, $\tilde{f}_i$ is not, and it changes for each $\mathcal{T}_i$. Moreover, $\tilde{f}_i$ grows, by construction, with the set index i [Blanchini and Miani, 2008]. Therefore, if an exact polyhedral ROSC set $\mathcal{T}_i$ is used, it is not possible to control the maximum value of $\tilde{f}_i$ as well as the number of operations that must be performed by the actuator.

In the next subsection, to limit the number of operations required on the actuator and keep it constant, we propose using zonotopic inner approximations of the ROSC sets $\{\tilde{\mathcal{T}}_i\}_{i=1}^N$.

4.2.2 Zonotopic approximation of the ROSC set family

In [Yang and Ozay, 2021], the authors proposed a methodology to construct a family of ROSC sets described by zonotopes. Such a solution has been shown to be able to significantly reduce the number of half-spaces $\tilde{f}_i$ required to represent $\tilde{\mathcal{T}}_i$. However, even adopting such a strategy, f_i still increases with the number of ROSC sets. To overcome such an undesired phenomenon, we propose a procedure to build zonotopic ROSC sets with a constant number of half-spaces. In particular, by exploiting the zonotopic inner approximation of polyhedral sets described in [Yang and Ozay, 2021, Sec. 4.A.2], the recursive definition of ROSC sets (17) can be modified as follows:

$$\mathcal{T}_i := \{x \in \mathcal{X} : \exists u \in \mathcal{U} \; s.t. \; Ax + Bu \in \tilde{\Xi}_{i-1}\}, \; i \geq 1 \tag{49}$$

where $\tilde{\Xi}_{i-1}$ is an inner zonotopic approximation of $\tilde{\mathcal{T}}_{i-1}$. Given a template Zonotope $Z(G, c)$ with a fixed number p of generators, the set $\tilde{\Xi}_{i-1} = Z(\gamma^* G, c^*)$ can be computed as follows [Yang and Ozay, 2021]:

$$\begin{gathered}[\gamma^*, c^*] = \arg\max_{\gamma, c} \sum_{i=1}^{p} d_i \log(\gamma_i), \;\; \text{s.t.} \\ H_i^{\tilde{\mathcal{T}}_i} c + |H_i^{\tilde{\mathcal{T}}_i} G| \gamma \leq g_i^{\tilde{\mathcal{T}}_i}, \; 0 \leq \gamma \leq 1\end{gathered} \tag{50}$$

where $d_i \geq 0$ are weighting factors and $|H_i^{\tilde{\mathcal{T}}_i} G|$ denotes a matrix obtained taking the element-wise absolute value of $H_i^{\tilde{\mathcal{T}}_i} G$. Similar to (17), the recursion (49) is stopped when $\mathcal{T}_{i-1} \not\subset \mathcal{T}_i$, or the domain of interest is covered.

4.2.3 E-ST-MPC properties

In what follows, we assume that E-ST-MPC operations described in section 4.2.1 are performed on a family of zonotopic ROSC sets $\{\tilde{\mathcal{T}}_i\}_{i=1}^N$ computed as prescribed in section 4.2.2, i.e., $\{\tilde{\mathcal{T}}_i\}_{i=1}^N \leftarrow \{\tilde{\Xi}_i\}_{i=1}^N$.

Proposition 4 *The E-ST-MPC scheme developed in sections 4.2.1 and 4.2.2 fulfills the objectives (O1)-(O3).*

Proof 4 *The proof can be obtained by collecting all the above developments:*

O1: The E-ST-MPC operations on the cloud are performed on the encrypted data without their decryption. Therefore, the assumed honest but curious cloud provider is not able to eavesdrop on the state measurements $x(k)$ and the computed control inputs $u(k)$.

O2: At each time step, E-ST-MPC determines the set-membership upper bound $\bar{i}(k)$ as in (39). As a consequence, the used one iteration of the PGM method, see (45)-(47), computes an admissible control input $u(k)$ that ensures that $x(k+1) \in \mathcal{T}_{\bar{i}(k)-1}$. As a consequence, in N steps, starting from any admissible $x(0)$ within the controller's domain, $x(N)$ is ensured to belong to $\mathcal{T}_0$. Consequently, also stability and recursive feasibility properties are fulfilled.

O3: Since the used family of ROSC set $\{\tilde{\mathcal{T}}_i\}$ is computed according to (49), then the number of half-spaces $\tilde{f}_i$ needed to describe each ROSC set is independent of i, and trivially upper bounded by 2^p, with p the number of generators. Moreover, if $p > n$, then $\tilde{f}_i \leq 2\binom{p}{n-1}$ [Zaslavsky, 1975]. As a consequence, given the maximum number of operations that the actuator can perform within the given sampling time, the maximum number of generators p that the control architecture can afford can be determined.

Remark 6 *If the matrix A is stored in the cloud in plaintext and the communication channel from the cloud to the controller has sufficient bandwidth to simultaneously support the transmission of $Enc[u^{0^+}]$ and an encrypted vector of dimension $\mathbb{R}^{\tilde{f}_i}$, then the computations required on the actuator's side can be further reduced. In particular, the sub-vector $\varepsilon = g^{\tilde{\mathcal{T}}}_{i^*(k)-1} - H^{\tilde{\mathcal{T}}}_{i^*(k)-1}Ax(k)$ in $g_i^{\mathcal{U}(x(k))}$ (see (43)) can be pre-computed encrypted on the cloud as follows:*

$$Enc[\varepsilon] = g^{\tilde{\mathcal{T}}}_{i^*(k)-1} \oplus \left[(-H^{\tilde{\mathcal{T}}}_{i^*(k)-1}A) \odot Enc[x(k)]\right] \tag{51}$$

Remark 7 *The possibility of controlling the number of operations performed on the actuator is unique in the related literature. For instance, the schemes in [Darup, 2020, Darup et al., 2017] limit the applicability of their solution to setup where the projection operator on the actuator can be efficiently performed, i.e., when the plant is subject to box-like input or ellipsoidal constraints. Therefore, in a more general context (where the plant is subject to arbitrary polyhedral state and input constraints) and where the projection (48) must be used, the time complexity of the operations on the actuator will be $\mathcal{O}(N_p(f_x + f_u))$, where N_p is the used prediction horizon that for feasibility reasons must be sufficiently large [Rawlings et al., 2017]. Our solution allows moving most of the required computations offline, leaving online the solution of an MPC problem with $N_p = 1$ and where the number of half-spaces representing the ROSC set can be imposed by design.* □

For the sake of completeness and to pave the way to possible extensions of this work, it is important to mention the limitations of the proposed solution. First, the inner approximation of the ROSC set might reduce the domain of attraction of the given controller. Therefore, the proposed solution presents a trade-off between the size of the controller's domain and the number of computations required on the actuator's side. Second, the use of the upper bound $\bar{\imath}(k)$ instead of $i^*(k)$

can increase, on average (with respect to a non-encrypted ST-MPC), the number of steps required to reach $\mathcal{T}_0$.

4.3 Simulation Results

To verify the effectiveness of the proposed solution, we consider the benchmark example of a two-tank water system. The encrypted controller operations have been simulated in Python where the library *eclib* (`https://github.com/KaoruTeranishi/EncryptedControl`) is used to implement the Paillier cryptosystem with $|p| = |q| = 1024$ bits. The states of the two-tank system are the level of water inside the two tanks, i.e. $x = [h_1, h_2]^T$, while the control input vector $u = [u_p, u_l, u_u]^T$ consists of the voltage of the pump u_p, and lower and upper valve signals u_l. By linearizing the system dynamics around the equilibrium pair $x_{eq} = [0.5, 0.5]^T, u_{eq} = [0.938, 1, 0.833]^T$ and by considering a sampling time $T_s = 1$ sec, the discrete-time model is given by $\tilde{x}(k+1) = A\tilde{x}(k) + B\tilde{u}(k) + d(k)$, where $\tilde{x} = x - x_{eq}, \tilde{u} = u - u_{eq}$, and

$$A = \begin{bmatrix} 0.993 & 0.003 \\ 0.007 & 0.982 \end{bmatrix}, B = \begin{bmatrix} 0.008 & -0.003 & -0.003 \\ 0.000 & 0.003 & 0.003 \end{bmatrix}$$

The following constraints and disturbance are prescribed: $-0.7778 \leq \tilde{u}_p \leq 0.6111$, $-1.25 \leq \tilde{u}_l \leq 0.75$, $-1.4765 \leq \tilde{u}_u \leq 0.5235$ and $-0.48 \leq \tilde{h}_1,\ \tilde{h}_2 \leq 0.3$, $\mathcal{D} = \{d \in \mathbb{R}^2 : |d_j| \leq 0.001,\ j = 1, 2\}$.

Moreover, we assume that the actuator's processor unit is capable of performing 12 iterations of the halfspace projection (48) within the given sampling time. Consequently, since $m = 3$ and the input constraints defines a box, i.e. $f_u = 2m = 6$, the E-ST-MPC is doable (i.e., the actuator can perform the required computations) as long as the number of generators used to builds the zonotpic inner approximations of the ROSC sets $\mathcal{T}_i$ is less or equal than 3. Indeed, if 3 generators are used, the number of required computations is upper bounded by $6 + 2\binom{3}{1} = 12$ (see O3 in the proof of Proposition 4). In the carried simulations, we have used as generators $v_1 = [-1.16, 1.058]^T$, $v_2 = [0, 0.226]^T$ and $v_3 = [0.5, 0.2]^T$. We have built a family of $N = 30$ ROSC sets (see the gray polyedra in Fig. 4.2), considered an initial state $\tilde{x}_0 = [-0.3,\ 0.29]^T \in \mathcal{T}_{29}$, and configured the PGM to use a step size of $\alpha = 0.4$. The ST-MPC and E-ST-MPC strategies are contrasted in Figs. 4.2-4.4. The obtained results show that despite the sub-optimality of the E-ST-MPC solution, the initial state $x(0)$ is driven in 30 steps to $\mathcal{T}_0$ (which is compatible with the theoretical upper proved in Proposition 4) by means of a sequence of admissible control inputs (see Fig. 4.4). Fig. 4.3 depicts how the set-membership index $\bar{i}(k)$ used by E-ST-MPC defines an upper bound of the actual set-membership signal $i(k)$ used by ST-MPC. The latter also explains why ST-MPC is able to drive the state trajectory into $\mathcal{T}_0$ using a lower number of steps.

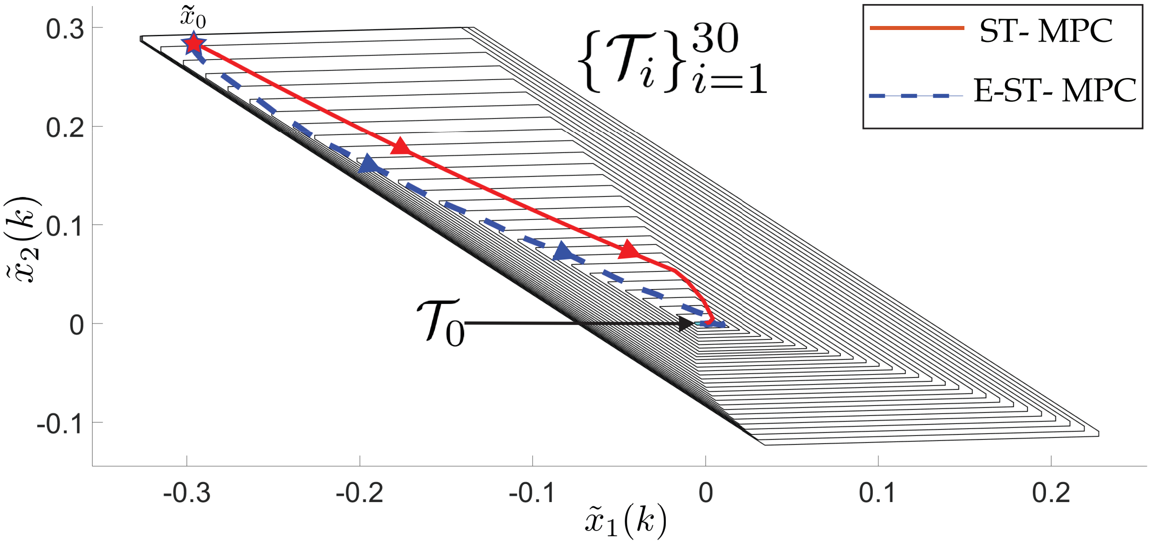


Figure 4.2: Computed family of ROSC sets and state trajectory.

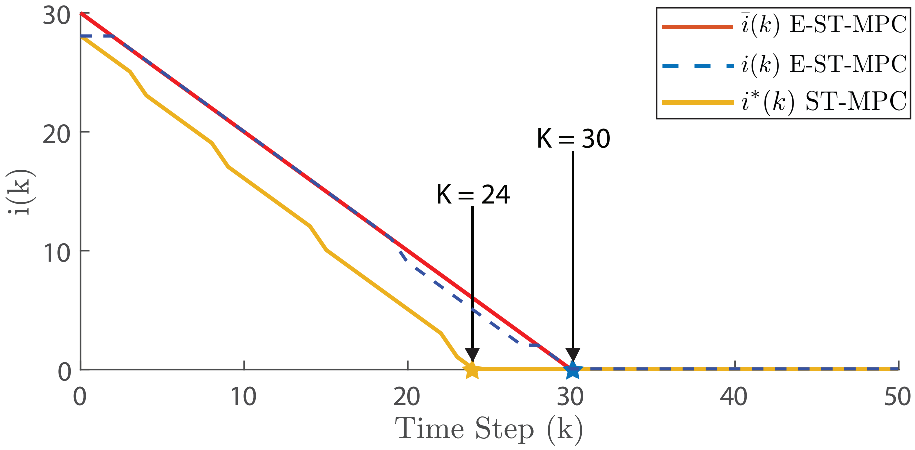


Figure 4.3: Set membership index.

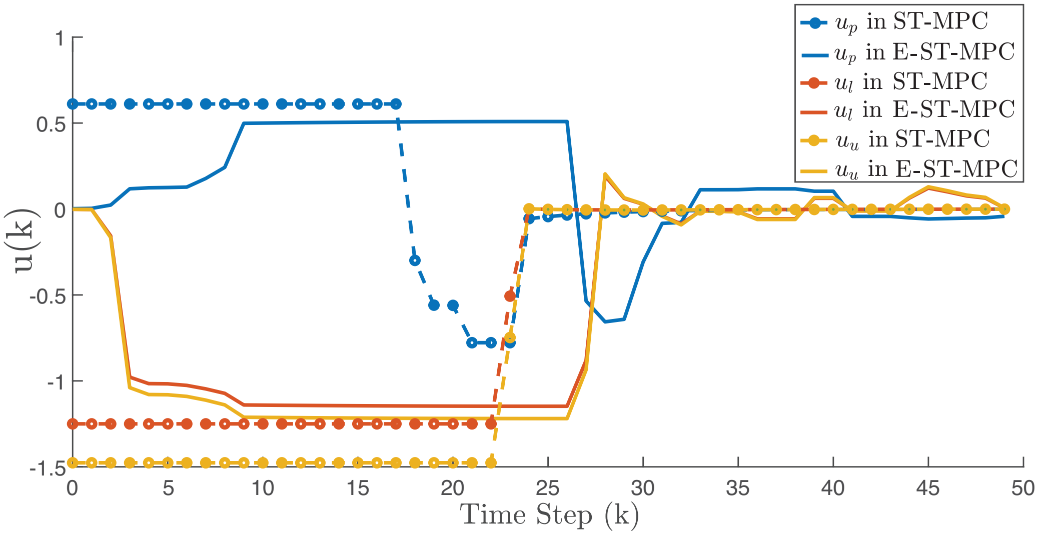


Figure 4.4: Control input signals.

4.4 Conclusions

We proposed an encrypted set-theoretic model predictive controller for single-cloud networked control systems. We also proved that all the properties of set-theoretic MPC control are preserved. Notably, the number of steps required to reach the smallest robust control invariant region is independent from the optimality of the encrypted solution. Unlike other existing approaches, the proposed control architecture does not present any computation trade-off between the prediction horizon and the number of computations on the actuator. Finally, the proposed approach has the unique feature of being able to control, at the design stage and independently from the prescribed constraints, the maximum number of plaintext operations that must be performed on the actuator. This has been obtained by taking advantage of zonotopic approximations of the used family of robust one-step controllable sets. Future works can focus on further improving the control performance in terms of the optimality of the solution and the conservativeness of the state set-membership estimation.

Chapter 5

On securing cloud-hosted cyber-physical systems using trusted execution environments

In this chapter, we propose a novel control architecture based on Trusted Execution Environments (TEE). We show that such an approach can potentially address major security and privacy issues for cloud-hosted control systems. Finally, we present an implementation setup based on Intel Software Guard Extensions (SGX), and validate its effectiveness on a testbed system.
The solution proposed in this chapter is published in the proceeding of the 2021 IEEE International Conference on Autonomous Systems (ICAS), see [Naseri et al., 2021]

5.1 System Setup and Threat Model

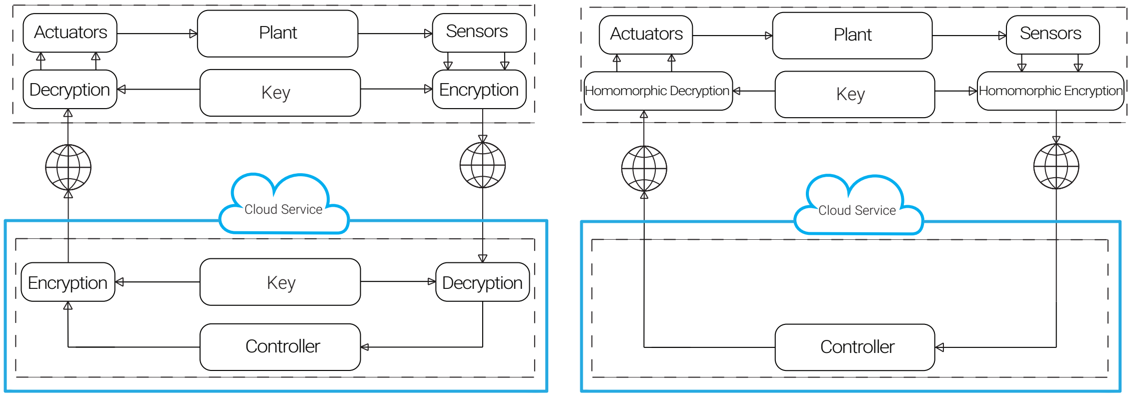
Threat Model. In this chapter, the following attacks, that can affect the privacy/security of the cloud-based CPS controllers, are considered.

Attacks against the communication channels - By adopting the conventional Dolev-Yao threat model [Dolev and Yao, 1983], a malicious entity with access to the public communication channels is assumed to be able to eavesdrop on the transmitted data and/or modify their content. Therefore, potentially, the confidentiality and integrity of the control system could be compromised. Indeed, such attackers can exploit the eavesdropped data to gain further information about the controlled system's behavior and use their disruptive capabilities to launch sophisticated undetectable attacks such as replay, covert, zero-dynamics attacks [Dibaji et al., 2019, Teixeira et al., 2015].

Attacks against the cloud service - If the cloud operator is malicious, or if the service is vulnerable, then an unauthorized entity (e.g., malware authors) might be able to gain access to the data transmitted between the plant and the controller, even if encrypted and authenticated communications are used. Indeed, such attackers could read the encryption key (key-management problem), intercept the transmitted data after decryption, and change the control logic (with the consequence of jeopardizing the whole control loop).

5.2 Existing Solutions

Different schemes have been proposed to secure networked control systems. A common solution is to use encrypted authenticated communications between the plant and the controller [Patel et al., 2009]; see Fig. 5.1a. Such a solution, at the cost of increased computational power to perform encryption/decryption operations at both the plant and controller's sides of the CPS, can mitigate the privacy and security issues related to cyber-attacks against the communication infrastructure. On the other hand, it does not address the security and privacy risks associated with the controller's deployment inside the cloud.



(a) Encrypted communications. (b) Homomorphic encryption.

Figure 5.1: Existing security solutions for cloud-based CPSs.

The use of homomorphic encryption has also been proposed to secure CPS solutions [Kogiso and Fujita, 2015, Farokhi et al., 2016]; see Fig 5.1b. A distinctive capability of such a solution is that it allows the controller to implement the control logic (in terms of additions and multiplication operations) directly on the received encrypted sensor measurements. Consequently, such an approach has the advantage of securing communications while solving the privacy issues associated with the cloud infrastructure. However, common drawbacks of homomorphic encryption include: the mathematical operations performed on the encrypted data are typically limited and computationally expensive, and the plaintext to ciphertext bit expansion factor is usually very high. Consequently, homomorphic-based solutions might not be practical for securing industrial control systems with fast sampling rates or narrow bandwidth.

There are three different types of homomorphic encryption schemes, namely partially homomorphic encryption (PHE), somewhat homomorphic encryption (SHE), and fully homomorphic encryption (FHE). Each subclass is characterized by the set and number of encrypted operations allowed. Therefore, according to the limitations imposed by the used scheme, it might be challenging to recast any existing control algorithm into its encrypted counterpart. For example, FHE allows an unlimited number of encrypted addition and multiplication operations. Therefore it is particularly appealing to implement sophisticated control solutions such as dynamic feedback control or model predictive control. However, such freedom comes with a computationally expensive bootstrapping process that makes FHE impractical to most control systems. Kim et al. [Kim et al., 2016] propose FHE to implement a dynamic output feedback controller using multiple controllers to avoid the bootstrapping delay. However, another inherent issue with FHE is that the ciphertext expansion might be up to $10000 : 1$ for an acceptable level of security of 100 bits [Chillotti et al., 2020]. Pailier's homomorphic encryption (PHE, supporting only encrypted additions) has also been proposed to implement a variety of controllers [Tran et al., 2020, Murguia et al., 2020]. However, due to memory issues related to the state of the dynamic encrypted controller (i.e., the number of bits required for its representation grows linearly with the number of iterations), the solution is usually limited to the use of resetting dynamics control laws. On the other hand, if a proportional controller is used, then, the control gain must satisfy some restrictive conditions imposed by the number of available bits [Lin et al., 2018].

Overall, existing solutions pose several limitations in terms of security/privacy/deployability to networked control systems. Moreover, no solutions have been proposed to protect CPSs against a malicious cloud operator, or malware that might be able to compromise the integrity of the control algorithm running on the cloud server.

5.3 Our Proposal

The objectives of our proposal are: secure the cloud-based CPSs against all the cyber-threat discussed in Section 5.1, and reduce the impact on the design and implementation of existing control strategies. The proposed secure control architecture has two essential components (see Fig. 5.2): an authenticated encryption scheme for securing the communication channels, and a TEE where the control logic is executed and the secret cryptographic keys, used by the authenticated encryption scheme, are stored.

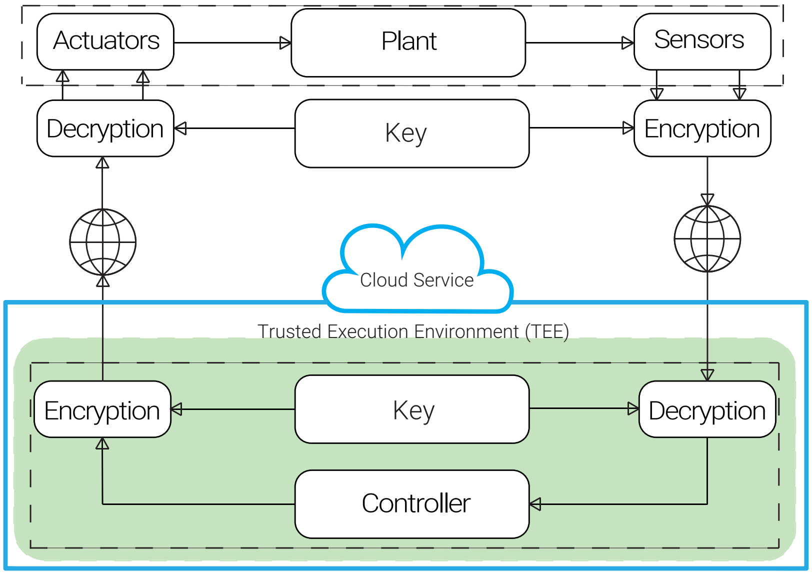


Figure 5.2: Proposed TEE-based solution.

First, we resort to authenticated encryption schemes (cf. [Patel et al., 2009]) to ensure the integrity and confidentiality of the control signal and sensor measurements exchanged between the plant and the controller. The used encryption scheme must be characterized by an inherent latency much smaller than the control-loop sampling time. The latter requirement is essential to ensure that the encryption scheme does not affect the control-loop system's stability. Second, a trusted execution environment (TEE) is used to protect the controller's operations in the cloud service. Generally speaking, a TEE refers to a hardware-based solution capable of ensuring that no malicious cloud entities (e.g., malware or a malicious cloud operator) could interfere with the execution of the control algorithm or with the memory associated with it. Moreover, if encryption/decryption operations are executed inside the TEE, where the keys are also protected by the TEE, then a malicious cloud administrator also cannot access the keys. TEE may also provide some other advantages such as measuring the integrity of the launched processes, measuring the origin of the TEE and the current state of the TEE (attestability), and recovering the state of the TEE to a known good state after any corruption (recoverability). The presence of a TEE on the plant side is not required for our threat model. However, it is desirable in a scenario where the local computing platform (e.g., SCADA system) could be subject to cyber-attacks. Several solutions have been proposed in the literature (not in CPS) using different TEE implementations, e.g., Intel SGX [Costan and Devadas, 2016], ARM TrustZone, AMD SEV [Kaplan et al., 2016], Hardware Security Module (HSM) [Varia et al., 2014], and secure co-processors [Bajaj and Sion, 2013]. Although all these solutions provide strong security mechanisms, not all can be used in our design (e.g., HSMs do not support remote attestation as opposed to Intel SGX).

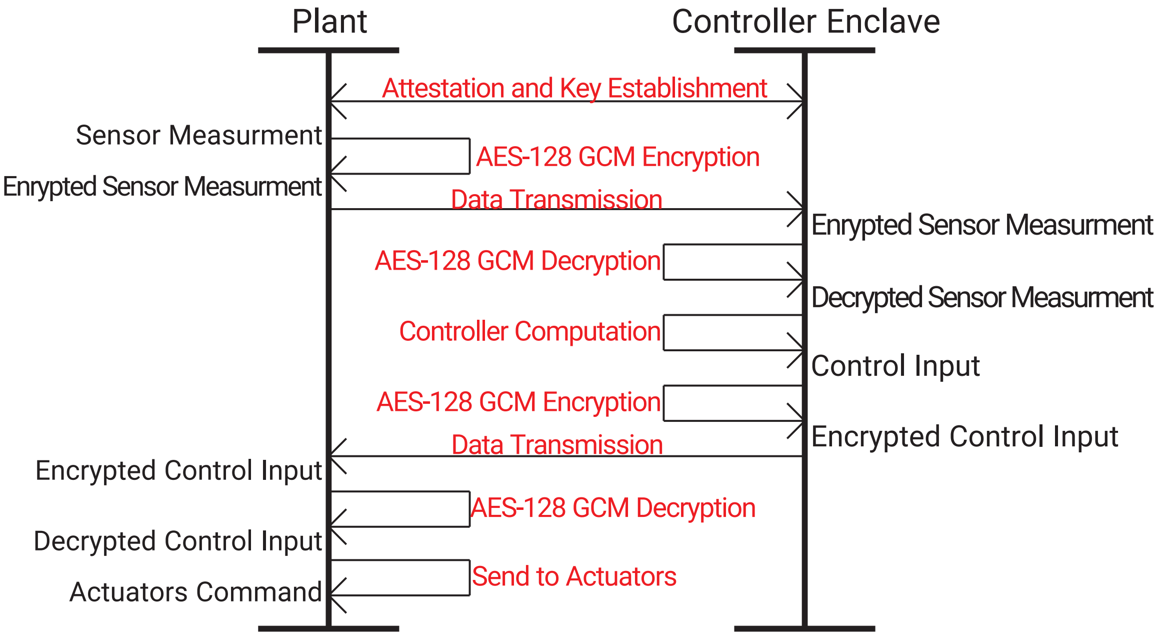


Figure 5.3: Data flow in the proposed solution.

5.4 Implementation

We use Intel SGX as TEE for its capability of providing a cryptographic attestation to ensure the integrity of the execution of the controller algorithm, even in the presence of a malicious cloud admin or a compromised cloud operating system (e.g., by malware). To keep code and data secure, SGX provides an isolated execution environment and encrypted memory. This secure container is called an "enclave" and everything else outside the enclave is assumed to be insecure. Two main functions are available to interact with the enclave, namely Enclave Call (E-CALL) and Out Call (O-CALL). E-CALL is used to call, from outside the enclave, a function implemented inside the enclave. On the other hand, O-CALL is used to call, from inside the enclave, a function implemented outside the enclave. For the implementation of the authenticated encryption, AES-128 Galois/Counter Mode(GCM) is used. This algorithm is a good candidate for CPSs because of its high throughput and low latency [Koteshwara et al., 2017, Arun et al., 2015]. First, we need to create an enclave and allocate memory for the Enclave Page Cache (EPC). The process starts with the attestation of both the enclave (validity of the CPU's SGX support) and the code (validity of the binary executed within the enclave as the controller logic). During the attestation, entities also establish a secure session key. After these initialization operations, data transmission will be started between the participating entities, encrypted under the session key. The data flow for a single control loop is shown in Fig. 5.3. In particular, the sensor measurements are encrypted on the plant side. Then, these encrypted sensor measurements are sent to the cloud over the communication channel. The authenticity of the received measurement is checked inside the enclave, where then the controller logic is also applied to the decrypted measurements. The evaluated controller output is then encrypted (inside the enclave) before it is sent to the actuator through the communication channel. Finally, the encrypted control input is decrypted by the actuator and applied to the plant.

5.5 Security and Performance Evaluation

We now discuss the security properties of the proposed solution. (i) *Confidentiality:* Data sent through the communication channels are encrypted with AES-128 GCM. Therefore, network eavesdroppers are unable to decrypt the transmitted control signals and sensor measurements. Moreover, control operations and encryption/decryption operations are performed within the enclave, avoiding the possibility that a malware or cloud administrator could intercept the plaintext signals or acquire the keys. (ii) *Integrity:* By exploiting the message authentication code (MAC) tag in AES-128 GCM, it is possible to verify the integrity of the transmitted data (i.e., detect if an attacker has manipulated the transmitted data). Another aspect of integrity is to make sure that the controller logic is not manipulated by the cloud provider before the code is executed within the enclave. For this purpose, an attestation operation is performed to make sure that the code executed in the enclave is exactly that is sent to the cloud service by the system admin. To improve code obfuscation (i.e., hiding the control logic from the cloud operator), the proposed solution in [Bauman et al., 2018] can be used. Note that the controller's runtime state remains always protected by SGX's memory encryption. Moreover, since the controller is executed inside SGX, the integrity of the control algorithm is also ensured. (iii) *Authentication:* The remote attestation feature of Intel SGX is used on the plant side to establish a secure and authenticated communication channel with the enclave in the cloud and ensure that the remote enclave is trusted. The MAC tags also is used by both entities (plant and controller) to make sure that the received messages are obtained by a trusted entity. (iv) *Freshness:* The uniqueness of the AES-128 GCM IV is used to guarantee the freshness of each message. Defending against side-channel attacks against Intel SGX [Brasser et al., 2017] is outside the scope of this paper. In the case of the necessity of storing data by the controller (depending on the controller logic), to mitigate rollback attacks on the sealed data, the Monotonic Counter (MC) of Intel SGX can be used to guarantee that the sealed data is the latest copy.

System setup. As a testbed, we use the Quadruple Tank Process (QTP) system from Johansson [Johansson, 2000], which is often used as a benchmark for control systems applications. The system consists of four water tanks where $h_i, i \in 1, 2, 3, 4$ represents the level of water in each tank and also represents the states x of the system, i.e., $x = [h_1,\, h_2,\, h_3,\, h_4]^2 \in \mathbb{R}^4$. There are two sensors that measure the level of water inside tanks 1 and 2, i.e., the output measurement vector is $y = [0.5h_1,\, 0.5h_2]^T \in \mathbb{R}^2$. Moreover, the system is equipped with two pumps, and the applied voltage v_1, v_2 are the inputs u of the system, i.e., $u = [v_1,\, v_2]^T$. We have linearized the system model around the equilibrium pair ($x_{eq} = [12.4,\, 12.7\,, 1.8,\, 1.4]^T$ $u_{eq} = [3,\, 3]^T$) and discretized it using a sampling time $T_s = 0.1\,\text{sec}$. The linearized model $x(k+1) = Ax(k) + Bu(k)$, $y(k) = Cx(k)$

and its matrices A, B, C can be easily obtained following [Johansson, 2000]. The plant is regulated by means of a dynamic output feedback controller consisting of a Luenberger Observer and an optimal Linear Quadratic (LQ) controller. The state-estimator operations are described by the discrete-time system $\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L(y(k) - C\hat{x}(k))$ where $\hat{x}(k)$ is the estimation of the state $x(k)$ and the correction gain is given by $L = \begin{bmatrix} 0.78 & 0 & 0.32 & 0 \\ 0 & 0.78 & 0 & 0.32 \end{bmatrix}^T$. The LQ controller logic is computed as $u = K(x - x_{eq}) + u_{eq}$ where the stabilizing gain is given by $K = \begin{bmatrix} 27.547 & -0.054 & 0.468 & 0.086 \\ 0.023 & 28.441 & 0.143 & 0.507 \end{bmatrix}$.

The dynamic output feedback controller operations have been implemented by utilizing an Intel SGX running on an Intel Core i7-6700 CPU, 3.40GHz, with 4 cores and 8 threads and 16 GB of RAM, using 64-bit Windows 7.

Measurements. We have conducted a series of measurements to evaluate the computation times required by different components of the proposed solution (see the data flow in Fig. 5.3). The reported CPU measurements have been obtained using the approach proposed in [Gjerdrum et al., 2017, Fig. 1], i.e., an O-CALL function is used as a stopwatch. As a result, the time measurements in Table 5.1 include an extra time representing the CPU time required to return to the enclave from an O-CALL and exit from it. We denote this time by Δt. To mitigate the presence of Δt in the measurements, we repeated each operation inside the enclave 1000 times and then calculate the average. Δt is also measured separately. The numerical results show that the two dominant factors are Δt and the control algorithm CPU time. Indeed, the average total CPU time required by both the secure and insecure implementations are around $905\mu s$ and $479\mu s$, respectively. The obtained results confirm that the computational overhead introduced by the use of Intel SGX does not affect the feasibility of the control strategy. Moreover, given that the introduced overhead is in the milliseconds' range, the proposed SGX-based secure architecture is believed to be affordable for a large class of cloud-based control systems applications.

Operation	Time (μs)
Enclave creation	8368.4
Dynamic output feedback controller	466.7
AES-128 GCM encryption	1.8
AES-128 GCM decryption	1.4
Δt	435.4

Table 5.1: Average time for different operations of the SGX-based solution

5.6 Conclusion

We proposed a solution to secure cloud-hosted/edge-hosted CPSs. In particular, by resorting to authenticated encryption and a trusted execution environment, we showed that the proposed networked control scheme is secure again different attacks against its security and privacy. We verified the effectiveness of such a scheme by means of numerical simulations obtained considering Intel SGX, where we performed different benchmarks to evaluate the computational burden associated with the trusted control scheme implementation. The obtained results show good promise in terms of real-time performance and simplicity of implementation in CPSs applications. The proposed solution can also be implemented in a non-cloud setting to help mitigate supply chain breaches.

Chapter 6

Confidentiality Attacks against Encrypted Control Systems

This chapter shows that encrypted control systems based on homomorphic encryptions are vulnerable to attackers leveraging the inherently small domains of the plaintext data in control systems and the randomization process required to make the utilized ciphers semantically secure. In particular, by considering the popular ElGamal and Paillier encryption schemes, we investigate different attacks that enable malware, which compromises the random number generator used by the randomized encryption schemes, to covertly leak the private decryption key and/or the measurements to an eavesdropper who has access to the measurement channel. Finally, we present some countermeasures to defend against these attacks.
The proposed attacks and countermeasures in this chapter are published in the journal of Cyber-Physical Systems (CPS), see [Naseri et al., 2022a].

6.1 Preliminaries and Problem Formulation

6.1.1 Encrypted Sampled-Data Networked Control System

Consider the encrypted NCS architecture shown in Fig. 6.1. In such a scheme, the plant is regulated by a networked controller implemented on a third-party platform (e.g., cloud), and accessible

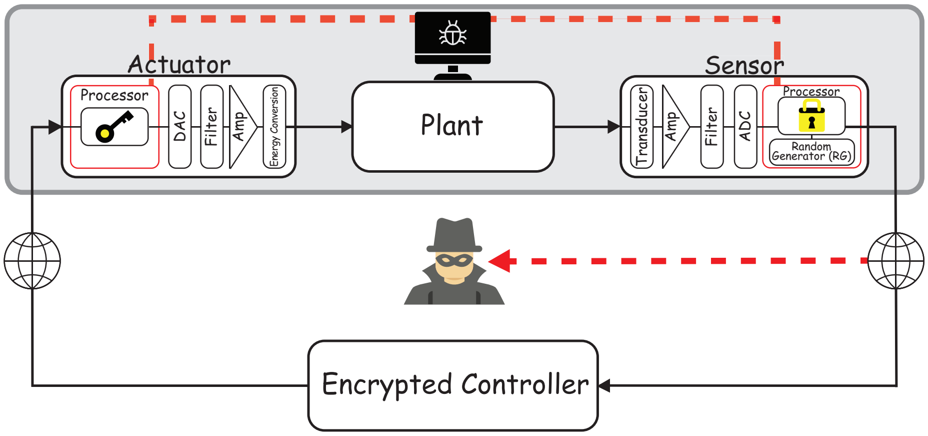


Figure 6.1: Encrypted control system using HE.

through a communication channel. To guarantee the confidentiality of the closed-loop control system (e.g., sensor measurements $y(t) \in \mathbb{R}^{n_p}$, $n_p \geq 1$, and control inputs $u(t) \in \mathbb{R}^{n_m}$, $n_m \geq 1$), the control-loop operates as follows. The sensor measurements $y(t)$ are encrypted into $Enc[y(t)]$ and then transmitted to the controller. The controller executes its logic directly on the received encrypted measurements $Enc[y(t)]$, producing in the output the encrypted control input vector $Enc[u(t)]$ (see the subsection 6.1.2 for more details) which is then sent to the actuators. A decryptor module, local to the actuator, recovers the plaintext control vector $u(t) = Dec[Enc[u(t)]]$, so allowing the actuator (by means of a digital-to-analog converter (DAC)) to apply $u(t)$ to the plant. Note that the "sensor" and "actuator" boxes in Fig. 6.1 are assumed to contain the operations performed by the sensor and actuator as well as the operations needed to support the used cryptosystem. For instance, since the encryption algorithms work on integer numbers, we assume that the sensor and actuator processor units are able to implement the required mapping function from fixed-point numbers to integers and vice-versa, see e.g., [Darup et al., 2021].

In what follows, for simplicity, we assume that the encryption operations of the $i{-}th$ component of $y(t)$, namely $Enc[y_i(t)]$, are performed on an implicit integer representation of $y_i(k)$. The fact that storing the encryption key on the controller side can endanger the privacy of the system (against either internal cloud adversary or external adversaries who may compromise the cloud service) implies that the encryption scheme used in Fig. 6.1 is not arbitrary, but it must belong to a class of homomorphic encryption schemes supporting a suitable set and number of mathematical operations on the encrypted data [Brakerski et al., 2014, Gentry, 2010, Paillier, 1999] (e.g., to implement the control logic). Hereafter, two popular schemes (used to implement encrypted control), namely "ElGamal" and "Paillier", will be considered. See Section 2.3 for a brief overview.

The control architecture in Fig. 6.1 defines a sampled-data control system where the signals transmitted over the network are digital while the ones applied to the plant are analog. This aspect is particularly relevant in an encrypted setup because it provides prior information on the maximum number of bits used to present/transmit digital data over the network. Also, all homomorphic encryption (HE) schemes proposed for use in cloud-based control systems work on integer message space as required by the used modulo arithmetic. Hence, to encrypt the $i-th$ component of the sensor measurement $y_i(t) \in \mathbb{R}^{n_p}$, $y_i(t)$ needs to be mapped onto the integer message space $\mathcal{M}$. The first step of the mapping is usually an element-wise approximation of the real-valued measurements with fixed-point numbers from the set $\mathbb{Q}_{\beta,\gamma,\delta} = \{-\beta^{\gamma}, -\beta^{\gamma} + \beta^{-\delta}, \ldots, \beta^{\gamma} - 2\beta^{-\delta}, \beta^{\gamma} - \beta^{-\delta}\}$, where $\beta \geq 1 \in \mathbb{N}$ is the basis and γ and $\delta \in \mathbb{N}$ are known as the magnitude and the resolution of the set, respectively. Such operations are described by the following mapping function:

$$g : \mathbb{R} \rightarrow \mathbb{Q}_{\beta,\gamma,\delta}$$

$$|g(y_i(t)) - y_i(t)| \leq \beta^{-\delta} \; \forall \; y_i(t) \in [-\beta^{\gamma}, \beta^{\gamma}]$$

Typically, the analog signals produced by the sensors are sampled and quantized into fixed-point numbers by the analog-to-digital converter (ADC) using, for each sensor measurement y_i, $i = 1, \ldots, n_p$ of $y(t)$, a finite small number of binary digits d. This procedure can be performed inside the sensor, by its processing unit. Given the fixed-point approximation $g(y_i(t))$ of $y_i(t)$, the next step prescribes a suitable mapping from $\mathbb{Q}_{\beta,\gamma,\delta}$ to the integer message space $\mathcal{M}$. This operation can be easily done by scaling $\mathbb{Q}_{\beta,\gamma,\delta}$ with the factor of β^{δ} in modulo φ where φ is a user-defined parameters for the used cryptosystem [Darup et al., 2021], i.e.,

$$(\beta^{\delta}\mathbb{Q}_{\beta,\gamma,\delta} \mod \varphi) \; \in \; \mathcal{M}$$

The processing unit of the sensor typically performs the above operation.

Remark 8 *As described in [Park et al., 2003, Ch. 5, Sec. 5.2.5], d is typically between 12 and 16. As a consequence, the size of the plaintext message's space $\mathcal{M}$, namely $|\mathcal{M}| = 2^d$ is relatively small. Besides, another limitation to the message space is imposed by the desire to compute the control logic directly on the encrypted variables, particularly in dynamic controllers [Murguia et al., 2020]. As explained in, e.g., [Cheon et al., 2018, Lin et al., 2018], according to the kind and number of mathematical operations required to compute the control action, the size of the plaintext variables should be sufficiently small to avoid overflow with the used modulo space.* □

6.1.2 Encrypted Controller

In this section, we recall how a simple static feedback controller in the form

$$u(t) = Ky(t), \quad K \in \mathbb{R}^{n_m \times n_p} \tag{52}$$

can be implemented in an encrypted fashion using El-Gamal and Paillier cryptosystems, see the survey paper [Darup et al., 2021] and references therein for a more detailed discussion. - *Encrypted control computation with El-Gamal*: Since El-Gamal is multiplicative homomorphic, the control law (52) can be computed in the encrypted domain if each sensor measurement $y_i(t)$ and each element K_{ij} of K are separately encrypted. Indeed, the controller can compute the following encrypted matrix $\Gamma(t)$,

$$\Gamma(t)=\begin{pmatrix} Enc[K_{11}]\otimes Enc[y_1] & \cdots & Enc[K_{1n_p}]\otimes Enc[y_{n_p}] \\ \vdots & \ddots & \vdots \\ Enc[K_{n_m 1}]\otimes Enc[y_1] & \cdots & Enc[K_{n_m n_p}]\otimes Enc[y_{n_p}] \end{pmatrix} \tag{53}$$

If $\Gamma(t)$ is transmitted to the actuator, then it can compute each component $u_i(t)$ of $u(t)$ as

$$u_i(t) = \sum_{j=1}^{n_p} Dec[\Gamma_{ij}(t)], \quad i = 1, \dots, n_m \tag{54}$$

- *Encrypted control computation with Paillier*: Since the Paillier cryptosystem is additively homomorphic, it is not possible to compute the matrix $\Gamma(t)$ as in (53). However, $\Gamma(t)$ can still be computed if each entry K_{ij} of K is in plaintext. Moreover, differently from El-Gamal, there is no need to transmit the entire matrix $\Gamma(t)$ to the actuator, because the summation required by (54) can be performed encrypted on the controller's side. Therefore, each $i-$th component of $u(t)$, $i = 1, \dots, n_m$, can be computed as:

$$Enc[u_i(t)] = (K_{i1} \odot Enc[y_1(t)]) \oplus \cdots \oplus (K_{in_p} \odot Enc[y_{n_p}(t)]) \tag{55}$$

6.1.3 Problem Formulation

Assumption 2 *(Threat Model) - The adversary model consists of two coordinated entities: (i) a malware capable of tampering with the RG module of the sensor's processing unit and, in one scenario, capable of accessing the private key stored in the actuator's processing unit on the plant's side of the NCS (e.g., by means of a supply chain attack [Yang et al., 2016]) and (ii) a passive eavesdrop capable of reading the encrypted sensor measurements $Enc[y(t)], \forall\, t$. It is important to note that we assume that no dedicated communication channels exist between the malware and the eavesdropper.*

The problem considered in this chapter can be summarized as follows: *Given the encrypted control architecture described in the Sections 6.1.1 - 6.1.2, show that under Assumption 2, an attacker is able to compromise the confidentiality of encrypted control systems by covertly revealing private information (e.g., secret encryption key or plaintext sensor measurements) to an eavesdropper intercepting the encrypted measurement channel.*

6.2 Proposed Attacks

In this section, under Assumption 2, three different attacks against the confidentiality of encrypted NCS are presented. In all these scenarios, the objective of the malware (the sender) is to covertly tamper the encryption operations to encode sensitive private information in the transmitted encrypted measurements $Enc[y(t)]$. On the other hand, the eavesdropper (the receiver), given the prior knowledge of the sender operations, has the objective to extract the embedded information from $Enc[y(t)]$ and reconstruct private data such as the plaintext sensor measurements $y(t)$ or the secret key $\mathcal{K}_{pr}$.

The considered attacks leverage two potential vulnerabilities of encrypted control systems, namely the *small size of the plaintext message space* $\mathcal{M}$ (see Remark 8) and the *randomness of the cryptosystems* (see Remark 2).

Attack Scenarios. According to the privileges that the malware can obtain, the following scenarios can be analyzed (see Fig. 6.1 for a better understating of the description below):

- SC_1- The malware is able to read the private key stored in the actuator's processing unit and repeat the calls for the random number generator and encryption operation, without outputting the ciphertext, until it satisfies a specific condition.
- SC_2- The malware is able to compromise the initial seed of the RG module.
- SC_3- The malware is able to map the output of the RG module into a restricted space (e.g., by setting some of the output bits of the RG module to zeroes or any pre-specified values).

In what follows, we explain the details of these attack scenarios.

Attack Scenario SC_1

Proposition 5 *Consider the encrypted NCS in Fig. 6.1. Under the scenario* SC_1*, the malware can covertly disclose the private key* $\mathcal{K}_{pr}$ *to the eavesdropper using the encrypted measurement channel.*

Proof - Given the assumed capabilities of the malware, at each time t, it can encode the $j-th$ bit of $\mathcal{K}_{pr}$, namely $\mathcal{K}_{pr}[j]$ into the parity bits of $Enc[y_i(t)]$, $i \in [1, \ldots, n_p]$. More precisely, the malware can re-compute the encrypted sensor measurement $Enc[y_i(t)]$ (with a different random number r) until the encrypted binary vector $Enc[y_i(t)]$ has a parity bit equals to $\mathcal{K}_{pr}[j]$. Such encoding operations are summarized in Algorithm 1.

On the other hand, the eavesdropper on the measurement channel can recover the transmitted secret key by simply sequentially storing the parity bit of the received encrypted sensor measurements. □

Remark 9 *Using Algorithm 1, the attacker is able to transmit, at each sampling time* t, n_p *bits of* $\mathcal{K}_{pr}$ *using a tampered but legitimate ciphertext that is indistinguishable from a normal ciphertext. For example, consider a case where the plant has two sensor measurements, i.e.,* $n_p = 2$*, the*

Algorithm 2 Encoding the binary secret

```
$\mathcal{K}_{pr}$ in the encrypted sensor measurements
Initialization: $length\_of\_secret = |\mathcal{K}_{pr}|, j = 0$
— $\forall\, t :$ —
if $j < length\_of\_secret$ then
    for $i = 1 : n_p$ do
        while (parity bit of $Enc[y_i(t)] \neq \mathcal{K}_{pr}[j]$) do
            $r \leftarrow$ generate a new random number $\in \mathcal{R}_{rg}$
            $Enc[y_i(t)] \leftarrow$ compute the encrypted sensor measurement $y_i(t)$
        end while
        $j = j + 1;$
    end for
end if
```

sampling time is $T_s = 1$ ms and the secret key $\mathcal{K}_{pr}$ is 1024 bits. In this setup, the attacker can embed 2 bits of $\mathcal{K}_{pr}$ at each sampling time t in the parity bit of $Enc[y_1(t)]$ and $Enc[y_2(t)]$. Therefore, after 512 *ms, the entire key is transmitted.*

Remark 10 *Note that the disclosure attack described in Propostion 5 leverages the randomness of the cryptosystem to launch the attack. As a consequence, this attack can be performed in both Paillier and El-Gamal. Moreover, by exploiting the same idea, the attacker can transmit any other sensitive information that the malware might have access to. For El-Gamal, it is implicitly assumed that each bit of $\mathcal{K}_{pr}$ is encoded in either the parity bit of c_1 or c_2.* □

Attack Scenario SC_2

Proposition 6 *Consider the encrypted NCS in Fig. 6.1. Under the scenario SC_2, if the malware and eavesdropped have offline shared a seed number ζ, then the malware can covertly enable the eavesdropper to correctly decode $Enc[y(t)]$.*

Proof - In SC_2, the malware can set the initial seed of the RG. As a consequence, the eavesdropper (who also knows ζ) can predict the entire sequence of random numbers r generated by RG. According to the used cryptosystem, the eavesdropper operations to recover y_i are as follows:

El-Gamal: According to (12), each scalar variable $y_i, i = 1, \ldots, n_p$ must be decrypted from $E[y_i(t)]$ as

$$y_i(t) = (c_1^{-\mathcal{K}_{pr}} \bmod p)(c_2 \bmod p)$$

However, since $c_1 = g^r \text{mod } p$ (see (11)) and $h = g^{\mathcal{K}_{pr}}$ (see (10)), we can re-write the above as

$$y_i(t) = \big(h^{-r} \bmod p\big)\,(c_2 \bmod p) \tag{56}$$

Therefore, since all the variables on the right-hand side of (56) are known, i.e., r, p, h with p, h part of the public key, to the eavesdropper, then $Enc[y_i(t)], i = 1, \ldots, n_p$ can be successfully recovered.

Paillier: According to (14),

$$Enc[y_i(t)] = (n+1)^{y_i(t)} r^n \bmod n^2$$

and, by exploiting the knowledge of r and of the public key n, we can multiply both sides by $(r^{-n} \bmod n^2)$, obtaining

$$Enc[y_i(t)]r^{-n} \bmod n^2 = (n+1)^{y_i(t)} \bmod n^2 \tag{57}$$

Then, by resorting to the binomial theorem and exploiting the mod operator (which makes zero all the terms of the binomial multiple of n^2), we can simplify the right-hand side of (57) and obtain

$$Enc[y_i(t)]r^{-n} \mod n^2 = (1 + ny_i(t)) \mod n^2$$

from which

$$y_i(t) = \frac{(Enc[y_i(t)]r^{-n} - 1)}{n} \bmod n^2 \tag{58}$$

concluding the proof. □

Note that in (58), the notation $\frac{a}{b}$ does not denote the modular multiplication of a times multiplicative inverse of b.; it denotes the quotient of a divided by b.

Attack Scenario SC_3

Proposition 7 *Consider the encrypted NCS in Fig. 6.1. Under the scenario SC_3, if the malware and the eavesdropper agree on a restricted random space $\mathcal{R}_{small} \subset \mathcal{R}$, then the malware can covertly enable the eavesdropper to correctly decode $Enc[y(t)]$.*

Proof - Given the knowledge of $\mathcal{K}_{pu}$, $Enc[y(t)]$ and $\mathcal{R}_{small}$, the eavesdropper, can perform the following actions:

El-Gamal: by taking advantage of the restricted random space $\mathcal{R}_{small}$ imposed by the malware, the eavesdropper can offline build a lookup table LT containing the following pairs:

$$\left\{ \left(r,\, \underbrace{g^r \bmod p}_{=c_1} \right) \,:\, r \in \mathcal{R}_{small} \right\}$$

As a consequence, given $Enc[y_i(t)] = (c_1, c_2)$, it is possible to use c_1 and LT to obtain r.

Paillier: by taking advantage of the restricted random space $\mathcal{R}_{small}$, given $Enc[y_i(t)]$, the eavesdropper can compute the set of admissible plaintext messages:

$$\mathcal{Y}(R_{small}) = \{ y_i(t) \in \mathcal{M} \,:\, r \in \mathcal{R}_{small},\, gcd(r,n) = 1,\, y_i(k) \text{ as in (58) } \} \tag{59}$$

Since the message space is restricted to $\mathcal{M}$, with $|\mathcal{M}| \ll |n|$ (see Remark 8), then the probability ρ of obtaining a random valid message $y_i \in \mathcal{M}$, given a randomly chosen $r \in \mathcal{R}_{small}$, is negligible, i.e., $\rho = \frac{1}{2^{|n|-|\mathcal{M}|}} \approx 0$. As a consequence, almost surely $\mathcal{Y}(\mathcal{R}_{small}) = y_i(t)$ (in a practical encrypted control setup, using Paillier, $\mathbb{Z}_n = \mathbb{Z}_{2^{1024}}$ and $\mathcal{M} = \mathbb{Z}_{2^{16}}$. Therefore, $\rho = \frac{1}{2^{1008}}$, and for restricted random space, e.g., $\mathcal{R}_{small} = \mathbb{Z}_{2^{32}}$, the cumulative probability that $\mathcal{Y}(\mathcal{R}_{small})$ contains two or more valid messages is practically zero). This concludes the proof.

From the above discussion, it follows that the time required by the attacker to recover y_i from its ciphertext in the case of El-Gamal is independent of $|R_{small}|$ since the eavesdropper is using a lookup table. However, for Paillier, this time grows exponentially with $|R_{small}|$. The important point, however, is that the eavesdropper finds only one admissible value for $y_i(t)$ in both cryptosystems, independent of $|R_{small}|$.

□

Remark 11 *In Paillier, if the more general form of the cryptosystem is used, i.e., $g \neq (n+1)$, then the attack in SC_2 and SC_3 can still be performed with some modifications:*

- SC_2: Since r is known and the message space is restricted, the eavesdropper can offline build a

lookup table $LT(\mathcal{M})$, containing the following pairs:

$$LT(\mathcal{M}) := \{(y_i, \underbrace{g^{y_i} r^n \mod n^2}_{=Enc(y_i)}) : y_i \in \mathcal{M}\}$$

Therefore, given $Enc[y_i(t)]$, the eavesdropper can obtain $y_i(t)$ from $LT(\mathcal{M})$.

- SC_3: By taking advantage of the restricted random and message spaces, the eavesdropper can offline build a lookup table $LT(\mathcal{M})$, containing the following pairs:

$$LT(\mathcal{M}) := \left\{\left(y_i, g^{y_i} \mod n^2\right) : y_i \in \mathcal{M}\right\}$$

Moreover, given $Enc[y_i(t)]$, the attacker can perform a search over the admissible random space ($r \in \mathcal{R}_{small}$, $gcd(r, n) = 1$) and compute $g^{y_i(t)} \mod n^2 = Enc[y_i(t)]r^{-n} \mod n^2$ until a value contained in $LT(\mathcal{M})$ is found. □

Remark 12 *In this section, we have developed the attacks in $SC1 - SC3$, assuming that the popular El-Gamal or Paillier cryptosystems are used. However, the proposed attacks leverage the inherently small domain of the message space $\mathcal{M}$ in control systems as well as the randomization process used in HE schemes. Therefore, the proposed attacks are valid for a more general class of encrypted control systems where the cryptosystem utilizes a randomization process for encryption.*

□

6.3 Countermeasures

Since the considered attacks exploit intrinsic vulnerabilities related to the random generator (RG) and small message space ($\mathcal{M}$)), existing anomaly/attack detectors for encrypted control systems (see e.g., [Kogiso, 2018, Baba et al., 2018, Teranishi and Kogiso, 2019] and references therein) are not effective. Moreover, this class of random generator attacks cannot be detected by analyzing the ciphertext (e.g., see [Austrin et al., 2014]). Consequently, instead of proposing an attack detection strategy, we hereafter introduce a solution that prevents their existence. Specifically, we

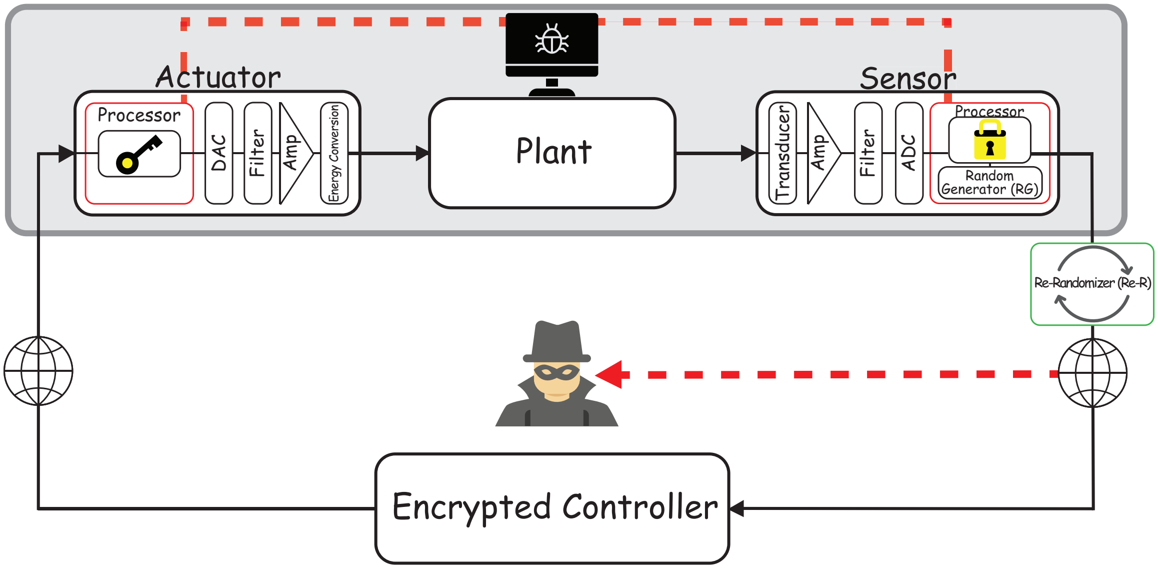


Figure 6.2: Encrypted control system with Re-Randomizer.

propose adding a new trusted subsystem, hereafter called "Re-Randomizer (Re-R)", between the sensor's subsystem and the communication channel (see Fig. 6.2), such that

(C_1) : Re-R takes in input $Enc[y_i]$, $\forall\, i$ and performs a re-randomizing of the ciphertext. By denoting with $\tilde{E}nc[y_i]$ the re-randomized version of $Enc[y_i]$, then y_i must be correctly decrypted from $\tilde{E}nc[y_i]$ by an entity possessing the private key $\mathcal{K}_{pr}$;

(C_2) : $\tilde{E}nc[y_i]$ prevents the attack scenarios SC_1-SC_3;

(C_3) : The Re-R's processor unit is completely independent of the actuator and sensor's units (i.e, if malware has access to the plant, then it cannot compromise Re-R).

Remark 13 *A possible way to ensure the independence and trustworthiness of Re-R is to implement its actions within a trusted execution environment [Costan and Devadas, 2016].*

The following proposition proposes possible re-randomization solutions for the El-Gamal and Paillier cryptosystems.

Proposition 8 *Consider a single encrypted message $Enc[y_i]$. In El-Gamal ($Enc[y_i] = (c_1, c_2)$), the re-randomization*

$$\tilde{E}nc[y_i] = (c_1 g^{\tilde{r}}, c_2 g^{\tilde{r}}), \quad \tilde{r} \in \mathcal{R}_{rg} \tag{60}$$

and in Paillier, the re-randomization

$$\begin{gathered}\tilde{E}nc[y_i] = Enc[y_i] \times \tilde{r}^n \bmod n^2, \\ \tilde{r} \in \mathcal{R}_{rg},\ \textit{s.t. } gcd(\tilde{r}, n) = 1\end{gathered} \tag{61}$$

fulfill the conditions (C_1)-(C_2).

Proof - The proof that (C_1) and (C_2) hold true is here split in two parts:

(C_1) : In El-Gamal, by construction, the encrypted message (60) is equal to

$$\tilde{E}nc[y_i] = (g^{r+\tilde{r}} \bmod p,\ y_i h^{r+\tilde{r}} \bmod p)$$

that, using (12) can be correctly decrypted into y_i. In Paillier, the encrypted message (61) is equal to

$$\tilde{E}nc[y_i] = (n+1)^{y_i}(r\tilde{r})^n \mod n^2$$

Moreover, since $gcd(r, n) = 1$ and $gcd(\tilde{r}, n) = 1$, then also $gcd(r\tilde{r}, n) = 1$. As a consequence, using (15), $\tilde{E}nc[y_i]$ can be correctly decrypted into y_i.

(C_2) : The re-randomization process randomly changes the parity bit of the encrypted variable $\tilde{E}nc[y_i]$. The latter is sufficient to nullify the attacker attempt in SC_1 to embed each bit of $\mathcal{K}_{pr}$ in the parity bit of $Enc[y_i]$, i.e., the probability of successfully decoding each bit of the $\mathcal{K}_{pr}$ is 0.5; Since the re-randomization embeds into the encrypted message $\tilde{E}nc[y_i]$ a new random number $\bar{r} \in \mathcal{R}_{rg}$ ($\bar{r} = r + \tilde{r}$ in El-Gamal, $\bar{r} = r\tilde{r}$ in Paillier), then the attacker is not aware of the used random number as well as it cannot restrict the random space. The latter is sufficient to conclude that the attack scenarios SC_2 and SC_3 are prevented. □

Therefore, the operations performed by the Re-R module can be summarized as follows:

(1) At each time-step t, the Re-Randomizer unit generates a new full-range random number $\tilde{r} \in \mathcal{R}_{rg}$. Moreover, if the Paillier cryptosystem is used, then $\tilde{r}$ must satisfy the condition $gcd(n, \tilde{r}) = 1$.

(2) Given $Enc[y_i(t)]$ and generated new random number $\tilde{r}$, the Re-R entity computes the re-randomized encrypted message $\tilde{E}nc[y_i(t)]$ according to the used cryptosystem:

- *El-Gamal:* $\tilde{Enc}[y_i(t)]$ is computed as in (60).
- *Paillier:* $\tilde{Enc}[y_i(t)]$ is computed as in (61).

(3) The re-randomized-encrypted messages $\tilde{Enc}[y_i(t)]$ are transmitted instead of $Enc[y_i(t)]$ to the controller.

6.4 Simulation Results

In this section, by considering a simple encrypted control system setup, we show the effectiveness of the attack scenarios described in section 6.2. The effectiveness of the proposed re-randomization technique is also verified. In the performed simulations, we considered a time-invariant discrete-time plant dynamical model whose state-space description is $x(t+1) = Ax(t) + Bu(t)$, $y(t) = Cx(t)$, and where

$$A = \begin{bmatrix} 1.01 & -0.01 \\ 0.00 & 1.02 \end{bmatrix}, B = \begin{bmatrix} 0.00 \\ 0.01 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$x(t) = y(t) \in \mathbb{R}^2, u(t) \in \mathbb{R}$. The plant is stabilized by a static feedback controller (52) where $K = [6.574, \ -6.201]$ and we assume that the ADC converter uses a sampling time period $T_s = 0.001\, s$ and $d = 16$ bits for the analog-to-digital conversion (see Remark 8) for each sensor measurement y_i, $i = 1, 2$. As a consequence, the considered message space is $|\mathcal{M}| = 2^{16}$. The El-Gamal and Paillier cryptosystems have been implemented with p, q such that $|p| = |q| = 1024$, and the encrypted control inputs are computed as in (54) and (55), accordingly. The encrypted control system operations have been simulated using the "eclib" python package[1]. By considering the attack scenario SC_1, Fig. 6.3, shows the number of key bits (of $\mathcal{K}_{pr}$) correctly recovered by the eavesdropper over time. The solid blue line depicts the result in the absence of the Re-R module, while the dashed red line in the case Re-R is used. In the absence of Re-R, the plot shows a slope equal to 1, denoting that all the bits are correctly decoded. On the other hand, the evolution of the red solid line shows that the eavesdropper can correctly decode (as expected) approximately 50% of the received key

[1] https://github.com/KaoruTeranishi/EncryptedControl

bits (see the proof of Proposition 8). Note that this does not provide the adversary with any useful information Since the adversary cannot know the positions of the correctly decoded bits.

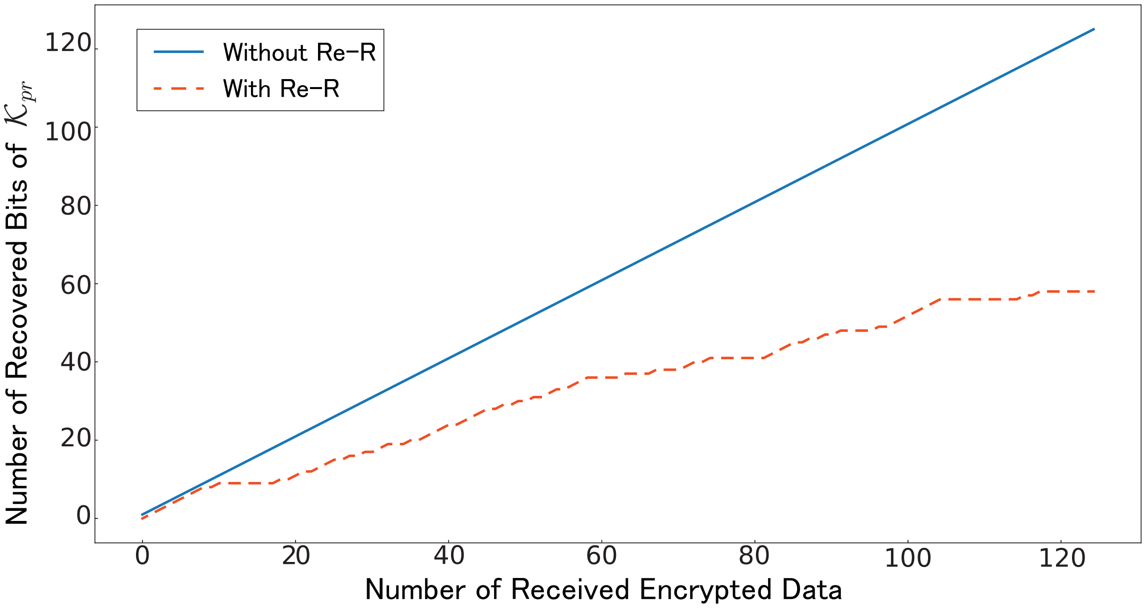


Figure 6.3: Number of recovered key bits in SC_1, with and without the re-randomization (Re-R) module.

By considering SC_2, and the Paillier cryptosystem, Fig. 6.4 shows, over a time interval of 2 seconds, the difference between the actual analog sensor measurements $y_i(t)$, $i = 1, 2$ and the decrypted value, namely $y_i^E(t)$, obtained by the eavesdropper using (58). The results show that the attacker can obtain $y_i(t)$ with an error that is limited only by the quantization error ($\frac{1}{2^{16}}$) in the considered ADC [Sokolov et al., 2019]. As a consequence, the attacker's estimation is identical to that obtained by the legitimate user using (15). Repeating the above experiment for SC3, produced an identical result to the one shown in Fig. 6.4.

Finally, Figs. 6.5 and 6.6 show the capability of the eavesdropper to correctly recover the sensor measurement data. In particular, for the time interval $[0,\,2]\,\mathrm{sec},$, both figures depict the sensor measurement data produced by the sensor and the values recovered by the eavesdropper in $SC2$ and $SC3$. It can be observed that the data recovered by the eavesdropper is equal to the quantized sensor measurements, which are encrypted and sent over the channel.

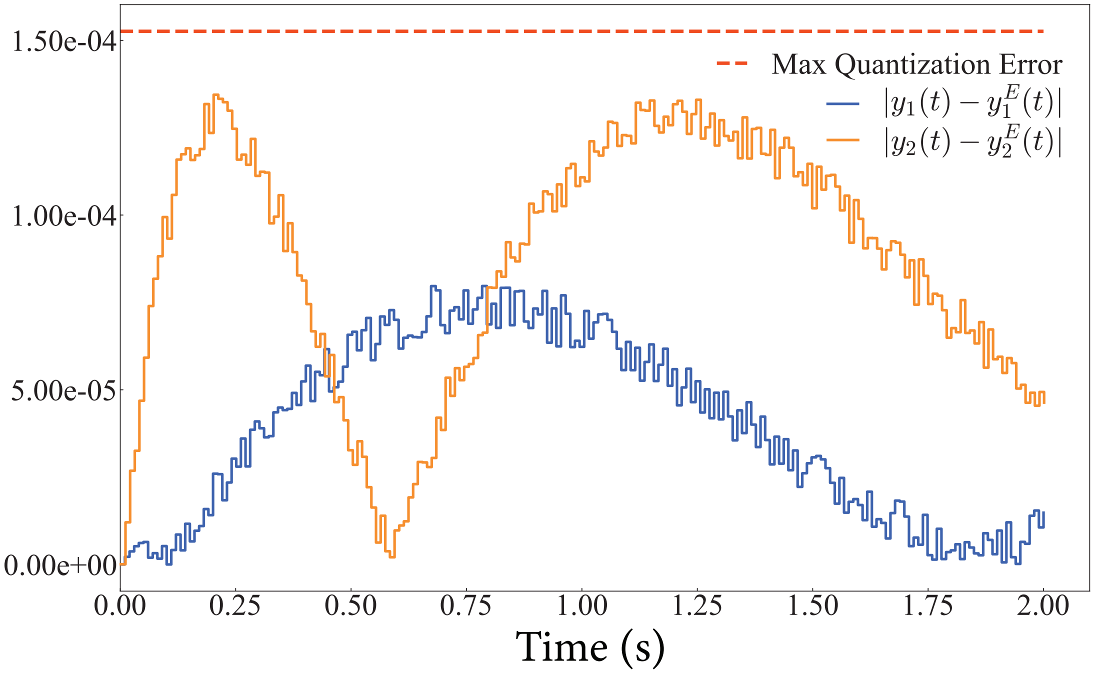


Figure 6.4: Difference between what the adversary can recover and the actual sensor measurements.

6.5 Conclusion

In this chapter, we have shown that different attacks can compromise the confidentially of encrypted control systems based on homomorphic cryptosystems. To the best of the authors' knowledge, no attacks against the confidentiality of encrypted control systems (e.g., sensor measurements, control inputs, controller parameters) have been reported in the CPS-related literature. In particular, we have shown that if an attacker is capable of deploying malware into the plant's side of the networked control system, then it can leverage intrinsic vulnerabilities (e.g., the limited message space and the randomness required to achieve semantic security of the encryption algorithms) to establish an illegitimate covert communication channel with an eavesdropper on the measurement channel. Then, we have proved that if a trusted re-randomization unit is used, these disclosure attacks are prevented.

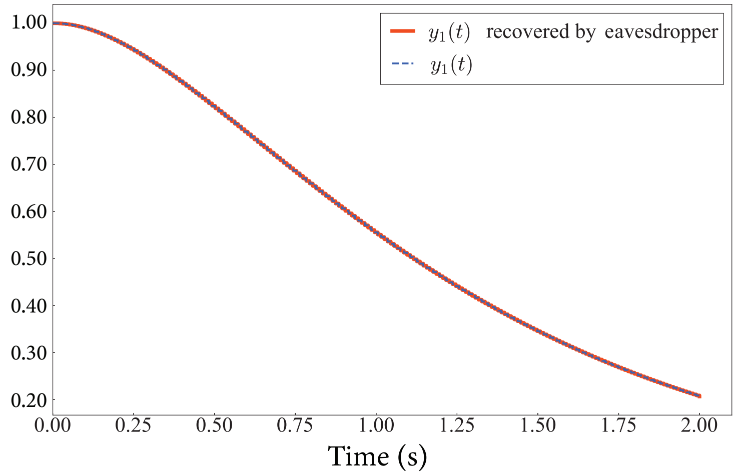


Figure 6.5: Attacker's capability to recover $y_1(t)$ in $SC2$ and $SC3$.

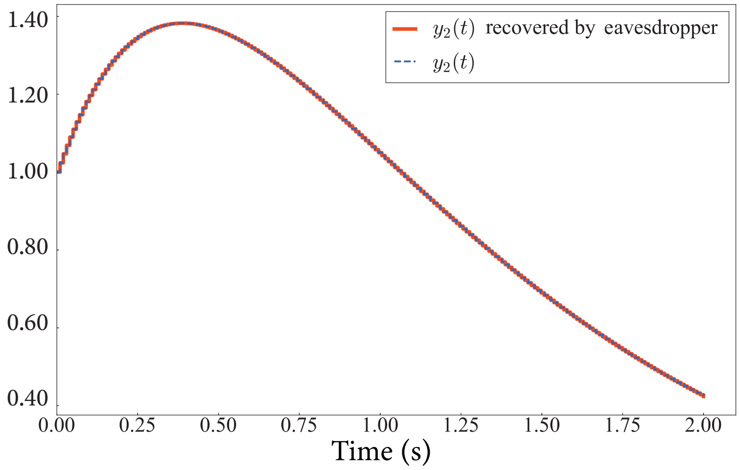


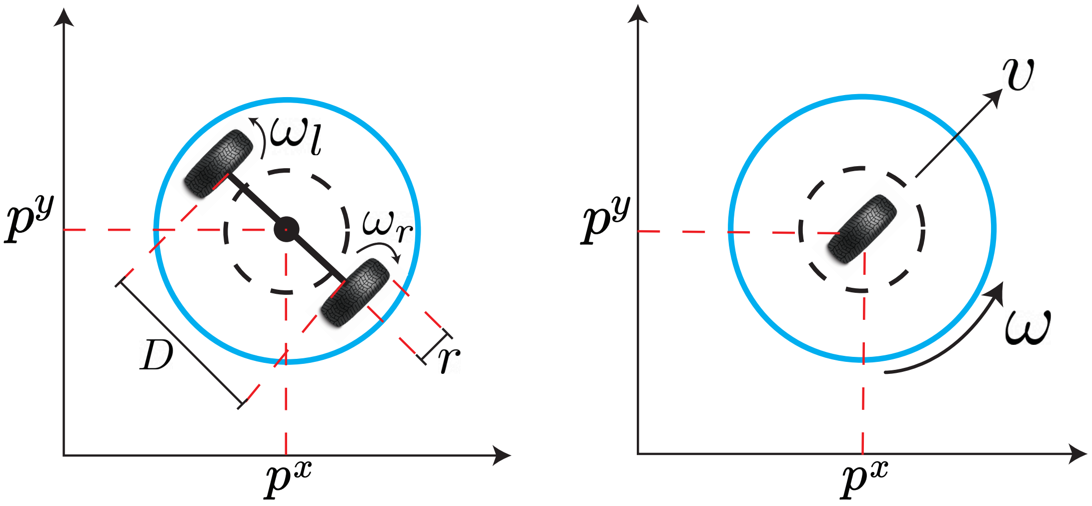
Figure 6.6: Attacker's capability to recover $y_2(t)$ in $SC2$ and $SC3$.

Chapter 7

An Observer-Based Key Agreement Scheme for Cyber-Physical Systems: The Differential-Drive Robot Case

Different control schemes that have been proposed to secure CPSs against sophisticated cyber-attacks require the exchange of secret messages between the plant and its remote controller. Thus, these schemes require pre-shared secret keys or an established Public Key Infrastructure (PKI) that allows for key agreement. In this chapter, we consider a control theoretic approach for establishing a secret key between the plant and the networked controller without resorting to traditional cryptographic techniques. Since remotely controlled mobile robots are a representative class of nonlinear Cyber-Physical Systems, the proposed protocol is developed considering the dynamics of a popular robot configuration known as a differential-drive robot. Our key agreement scheme leverages a nonlinear unknown input observer and an error correction code mechanism to allow the robot to securely agree on a secret key with its remote controller. To validate the proposed scheme, we implement it using a Khepera-IV robot and evaluate its efficiency and the additional control cost acquired by it. Our experimental results confirm the effectiveness of the proposed key establishment scheme.

The proposed key agreement in this chapter and the experimental results are submitted in the proceeding of the 22nd world congress of the International Federation of Automatic Control (IFAC) in 2023 and it is under review. A demo of the performed experiment is available at the following weblink `https://youtu.be/9FJkQhj8sdY`.



(a) Differential Drive. (b) Unicycle.

Figure 7.1: Differential-drive and unicycle models.

7.1 Preliminaries and Problem Formulation

Definition 8 *Given three positive integers $n_c \in \mathbb{Z}_{>0}, k_c \in \mathbb{Z}_{>0}, d_c \in \mathbb{Z}_{>0}$, a linear Error Correcting Code (ECC) defines a linear transformation of a binary string $s \in \{0,1\}^{k_c}$ into a subspace $\mathcal{C} \in \{0,1\}^{n_c}$ of cardinality 2^{k_c} such that*

- *$\forall(c_1, c_2) \in \mathcal{C}, c_1 \neq c_2$, the Hamming distance $d_H(c_1, c_2) < d_c$.*
- *the maximum number of errors that can be corrected is $\frac{d_c - 1}{2}$.*

□

In what follows, we consider a scenario where a mobile robot is maneuvered by a networked controller and the network infrastructure is vulnerable to eavesdropping attacks.

7.1.1 Robot Model

Among different existing categories of mobile robots, wheeled-mobile robots are very common for ground vehicles and they find application in different domains such as surveillance and warehouse automation. Moreover, among the nonholonomic configurations, the differential-drive structure, characterized by two rear independently-driven wheels and one or more front castor wheels for body support, is often adopted in the industry [Martins et al., 2017]. A schematic of a differential-drive robot is shown in Fig. 7.1a.

The pose of a differential-drive robot is described by the planar coordinates (p^x, p^y) of its center of mass and orientation θ (see Fig. 7.1a). By resorting to the forward Euler discretization method and a sampling time $T > 0$, the discrete-time kinematic model of the differential-drive is given by [De Luca et al., 2001]:

$$\begin{aligned} p^x(k+1) =& \quad p^x(k) + \tfrac{Tr}{2}\cos\theta(k)(\omega_r(k) + \omega_l(k)) + \zeta^{p^x}(k) \\ p^y(k+1) =& \quad p^y(k) + \tfrac{Tr}{2}\sin\theta(k)(\omega_r(k) + \omega_l(k)) + \zeta^{p^y}(k) \\ \theta(k+1) =& \quad \theta(k) + \tfrac{Tr}{D}(\omega_r(k) - \omega_l(k)) + \zeta^{\theta}(k) \end{aligned} \tag{62}$$

where $r > 0$ is the radius of the wheels, $D > 0$ the rear axle length, and $u^D = [\omega_r, \omega_l]^T \in \mathbb{R}^2$ the control input vector, which consists of the angular velocities of the right and left wheel, respectively. $\zeta(k) = [\zeta^{p^x}(k), \zeta^{p^y}(k), \zeta^{\theta}(k)]^T \sim \mathcal{N}(0, \mathcal{W})$ is the process noise with $\mathcal{W} \in \mathbb{R}^{3\times 3}$. Let $x(k) = [p^x(k), p^y(k), \theta(k)]^T \in \mathbb{R}^3$ denote the robot's state vector. It is assumed that $x(k)$ can be estimated leveraging the measurement vector $y(k) \in \mathbb{R}^{n_p}$, $n_p > 0$, obtained via odometric calculations and/or exteroceptive (e.g., sonar, laser) sensors [D'Alfonso et al., 2015], i.e.,

$$y(k) = h(x(k)) + \xi(k) \tag{63}$$

where $h(x(k))$ denotes the nonlinear output equation, and $\xi(k) \sim (0, \mathcal{V})$, $\mathcal{V} \in \mathbb{R}^{n_p \times n_p}$, the measurement noise, uncorrelated with $\zeta(k)$.

By denoting with $v(k)$ and $\omega(k)$ the linear and angular velocities of the center of mass of the robot, it is possible to apply to (62) the transformation

$$\begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix} = H \begin{bmatrix} \omega_r(k) \\ \omega_l(k) \end{bmatrix}, \quad H := \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{D} & \frac{-r}{D} \end{bmatrix} \tag{64}$$

and describe the robot's behavior by means of the following unicycle model (see Fig. 7.1b):

$$\begin{aligned} p^x(k+1) = &\ p^x(k) + Tv(k)\cos\theta(k) + \zeta^{p^x}(k) \\ p^y(k+1) = &\ p^y(k) + Tv(k)\sin\theta(k) + \zeta^{p^y}(k) \\ \theta(k+1) = &\ \theta(k) + T\omega(k) + \zeta^{\theta}(k) \end{aligned} \tag{65}$$

where $u^U(k) = [v(k), \omega(k)]^T \in \mathbb{R}^2$ is the control input vector of the unicycle.

7.1.2 Adversary Model

We assume a passive adversary capable of eavesdropping on the control input and sensor measurements transmitted between the plant and the networked controller, see Eve in Fig. 7.2. We also assume that the adversary is aware that the robot is a differential-drive robot but it might not have exact knowledge of all the robot's parameters (e.g., $T, r, D, \mathcal{W}$) and robot's measurement function (e.g., $h(\cdot)$ and $\mathcal{V}$). Therefore, we assume that the adversary has the following model:

$$\begin{aligned} p_a^x(k+1) = &\ p^x(k) + \tfrac{T_a r_a}{2}\cos\theta_a(k)(\omega_r(k)+\omega_l(k)) + \zeta_a^{p^x}(k) \\ p_a^y(k+1) = &\ p^y(k) + \tfrac{T_a r_a}{2}\sin\theta_a(k)(\omega_r(k)+\omega_l(k)) + \zeta_a^{p^y}(k) \\ \theta_a(k+1) = &\ \theta_a(k) + \tfrac{T_a r_a}{D_a}(\omega_r(k)-\omega_l(k)) + \zeta_a^{\theta_a}(k) \\ y_a(k) = &\ h_a(x_a(k)) + \xi_a(k) \end{aligned} \tag{66}$$

where $\zeta_a = [\zeta_a^{p^x}(k), \zeta_a^{p^y}(k), \zeta_a^{\theta}(k)]^T \sim (0, \mathcal{W}_a)$, $\xi_a^x(k) \sim (0, \mathcal{V}_a)$, and $(T_a, r_a, d_a, h_a(\cdot), \mathcal{W}_a, \mathcal{V}_a)$ are the adversary estimations for the robot's model (62)-(63).

Assumption 3 *Let $\mathcal{M} = \{T, r, D, \mathcal{W}, h(\cdot), \mathcal{V}\}$ and $\mathcal{M}_a = \{T_a, r_a, D_a, \mathcal{W}_a, h_a(\cdot), \mathcal{V}_a\}$ be the robot's model knowledge available to the controller's designer and to the adversary, respectively.*

Then,

$$\mathcal{M} \neq \mathcal{M}_a \tag{67}$$

Remark 14 *The model discrepancy (67) might arise for different reasons. First, the adversary might not be aware of the robot construction parameters r, D or the output function $h(\cdot)$. Instead, the attacker might just be able to estimate them using identification techniques or by inspection (e.g., via cameras). Second, while the defender can estimate $\mathcal{W}, \mathcal{V}$ by performing offline experiments, see, e.g., [Antonelli and Chiaverini, 2007, D'Alfonso et al., 2015], the eavesdropper can only perform online identification procedure relying on the online robot operations, which might be unsuitable for system identification purposes.*

7.1.3 Problem Formulation

The here considered key-agreement problem can be stated as follows.

Problem 1 *Consider the robot and adversary models (62)-(67). Without resorting to traditional cryptographic schemes, design a key agreement protocol between the robot and the networked controller such that the keys of length $n > 0$ identified by the controller ($\mathcal{K}_c \in \{0,1\}^n$), robot ($\mathcal{K}_r \in \{0,1\}^n$) and attacker ($\mathcal{K}_a \in \{0,1\}^n$) are such that*

$$P\{\mathcal{K}_c = \mathcal{K}_r\} \approx 1 \text{ and } P\{\mathcal{K}_c \neq \mathcal{K}_a\} \approx 1 \tag{68}$$

7.2 Key Agreement Protocol

As proved in [Lucia and Youssef, 2020], the asymmetry (67) in the plant model knowledge is sufficient to ensure the existence of a Wyner wiretap-like channel in networked cyber-physical systems. The latter is here leveraged to design an encoding mechanism for the considered key-exchange problem. In particular, the proposed key agreement protocol is developed under the following assumptions.

Assumption 4 *The available sensor measurements are sufficiently rich to allow the existence of an Unknown Input Observer (UIO) capable of simultaneously estimating $x(k)$ and $u^D(k)$ from the*

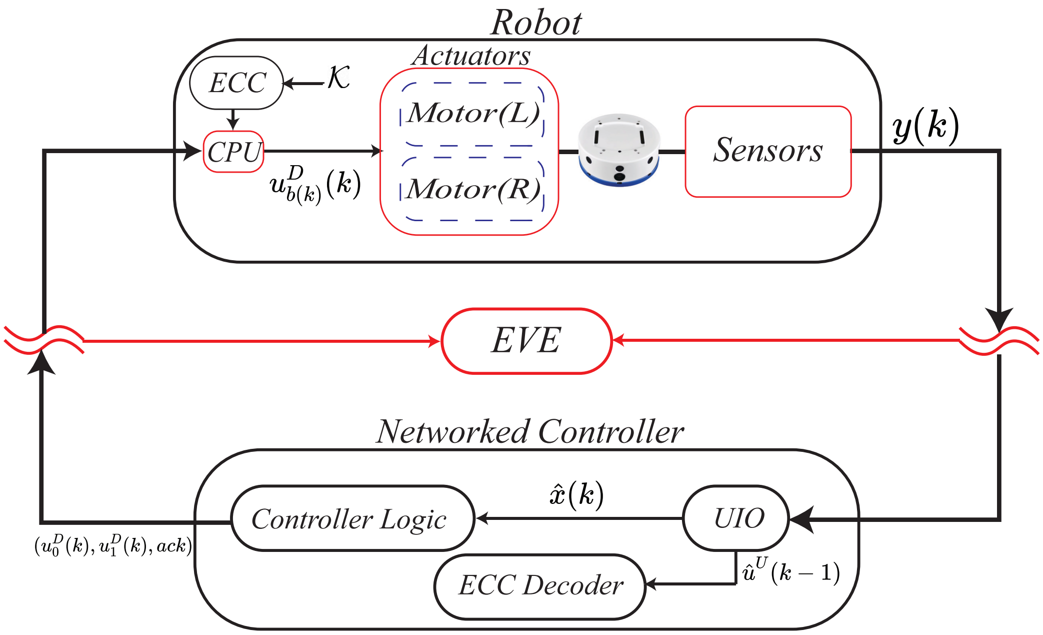


Figure 7.2: Control architecture for the proposed key agreement protocol.

available measurement vector $y(k)$. By denoting with $\hat{x}(k)$ and $\hat{u}^D(k)$ the estimated vectors, the UIO is abstractly modeled as the following recursive function

$$[\hat{u}^D(k-1), \hat{x}(k)] = UIO(u^D(k), \hat{x}(k-1), y(k), \mathcal{M}) \tag{69}$$

where $(\hat{u}^D(k-1), \hat{x}(k))$ and $(\hat{u}^D(k-2), \hat{x}(k-1))$ are the available estimations at time steps k and $k-1$, respectively. Moreover, the eavesdropper is able to run the same UIO as in (69) with $\mathcal{M}_a$ instead of $\mathcal{M}$.

In the sequel, we assume that the robot is equipped with a tracking controller which provides the control vector $u^U(k), \forall\, k$, i.e.,

$$u^U(k) = \begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix} = f_c(x(k), x_r(k), \dot{x}_r(k), \ddot{x}_r(k)) \tag{70}$$

where $f_c(\cdot,\cdot,\cdot)$ denotes a generic controller, $x_r(k) \in \mathbb{R}^3, \dot{x}_r(k) \in \mathbb{R}^3, \ddot{x}_r(k) \in \mathbb{R}^3$ are the reference state, velocity, and acceleration vectors, respectively [De Luca et al., 2001].

By referring to the networked control system architecture illustrated in Fig. 7.2, the idea behind the proposed key-agreement protocol can be described in four points:

(P1) - The controller computes $u^U(k)$ as in (70). Then, it generates two perturbed control inputs, namely $u_0^U(k) \in \mathbb{R}^2$ and $u_1^U(k) \in \mathbb{R}^2$, by adding and subtracting a small bias vector $\Delta \in \mathbb{R}^2$ to $u^U(k)$, i.e.,

$$u_0^U(k) = u^U(k) + \Delta, \quad u_1^U(k) = u^U(k) - \Delta \tag{71}$$

where $\Delta = [\Delta_v, \Delta_\omega]^T$, $\Delta_v \geq 0$, $\Delta_\omega \geq 0$ and such that $\Delta_v + \Delta_\omega > 0$ (i.e., at least one between Δ_v and Δ_ω must be strictly greater than zero). Finally, the differential-drive control inputs are computed as $u_0^D(k) = H^{-1}u_0^U(k)$ and $u_1^D(k) = H^{-1}u_1^U(k)$, see (71), and the pair $(u_0^D(k), u_1^D(k))$ is sent to the robot.

(P2) - Once the robot receives $(u_0^D(k), u_1^D(k))$, its CPU unit is in charge of deciding which one of the two control inputs should be used. To this end, it generates a random bit $b(k) \in \{0, 1\}$ and send to the actuators $u_{b(k)}^D(k)$. Note that the bit $b(k)$ and, consequently, the control signal applied to the robot $(u_{b(k)}^D(k))$ are unknown to the networked controller and to the eavesdropper. At each iteration, the robot appends $b(k)$ to the local key $\mathcal{K}_r$.

(P3) - When the networked controller receives $y(k)$, it can run the UIO (69) and obtain the estimated pair $(\hat{x}(k), \hat{u}^D(k-1))$. Moreover, since also the pair $(u_0^D(k-1), u_1^D(k-1))$ is known, the controller can estimate the random bit $b(k-1)$ (used by the robot) as

$$\hat{b}(k-1) = \begin{cases} 0 & \textit{if } d_0 < d_1 \\ 1 & \textit{if } d_1 < d_0 \end{cases} \tag{72}$$

where d_0 and d_1 are the distances between the estimated control input $\hat{u}^D(k-1)$ and $(u_0^D(k-1), u_1^D(k-1))$, i.e.,

$$\begin{aligned} d_0(k-1) &= \|\hat{u}^D(k-1) - u_0^D(k-1)\|_2, \\ d_1(k-1) &= \|\hat{u}^D(k-1) - u_1^D(k-1)\|_2 \end{aligned} \tag{73}$$

At each iteration, the networked controller appends $\hat{b}(k)$ to the local key $\mathcal{K}_c$.

(P4) - The adversary can run the UIO (69) with $\mathcal{M}_a$ instead of $\mathcal{M}$ and obtain a local estimation, namely $\hat{b}_a(k-1)$, of $b(k-1)$, to append to its local key $\mathcal{K}_a$. However, given the model discrepancy

(67), the covariance of the unknown input estimation error for the attacker is expected to be larger than the one obtained by the networked controller [Lucia and Youssef, 2022]. Consequently, for a proper choice of Δ, it is expected that $P\{\mathcal{K}_c = \mathcal{K}_r\} \approx 1$ and $P\{\mathcal{K}_c \neq \mathcal{K}_a\} \approx 1$.

Note that the above-described UIO-based decoding scheme might not be robust against possible model mismatches and/or process and measurement noises. To make the protocol more robust, we enhance its decoding operations by means of an Error Correcting Code (ECC) scheme and a feedback acknowledgment signal, namely ack, which is sent by the controller along with the pair of control inputs.

By assuming, for the sake of simplicity and clarity, a linear ECC, the ECC and ack feedback signal are used as follows (refer to Definition 8 for the used notation and terminology):

- The robot splits a randomly generated local key $\mathcal{K}$ into a sequence of substring s_i. Each s_i is encoded into a sequence of codewords c_i. Each bit of c_i, namely $c_i[j]$, is sequentially used to decide $b(k)$ in *(P2)*, i.e., $b(k) = c_i[j]$.

- The robot estimates $\hat{b}(k)$ as in *(P3)* and collects them to obtain an estimation of the codewords c_i, namely $\hat{c}_i$. Then, the Hamming distance $d_{\hat{c}_i}$ is evaluated

$$d_{\hat{c}_i} = \arg\min_{c \in \mathcal{C}} d_H(c, c_i) \tag{74}$$

  If $d_{\hat{c}_i}$ is much smaller than the number of correctable errors, then the codeword is accepted, and the binary string $\hat{s}_i$ (associated to $\hat{c}_i$ via ECC) is appended $\mathcal{K}_c$, and a positive $ack_i = 1$ is sent. Otherwise, the codeword is discarded and $ack_i = 0$ is sent.

- The robot, for every received $ack_i = 1$, append c_i to $\mathcal{K}_r$.

The complete key-agreement protocol is summarized in Algorithm 3.

Remark 15 *The bias Δ in (71) and the ECC parameters (n_c, k_c, d_c) are design parameters that can be tuned to achieve $P\{\mathcal{K}_c = \mathcal{K}_r\} \approx 1$. Moreover, to ensure the correctness of the exchanged key, the controller and the robot can always publicity verify its correctness by exchanging the hash values associated with $\mathcal{K}_c$ and $\mathcal{K}_r$. Moreover, to eliminate the partial key knowledge gained by the*

Algorithm 3 Proposed Key Agreement Protocol

```
− − − − − − − − − − − − − − − − − − − − $ROBOT$ − − − − − − − − − − − − − − − − − − −
Initialization: Generate $\mathcal{K}$, and set $\mathcal{K}_r = \emptyset$.
Split $\mathcal{K}$ into sub-strings $s_i \in \{0,1\}^{k_c}$
Sequentially encode each $s_i$ is into codewords $c_i \in \{0,1\}^{n_c} \in \mathcal{C}$
loop    At each time step $k$:
    Sequentially use each bit $c_i[j]$ of $c_i$ to pick $b(k) = c_i[j]$ and apply to the robot $u^D_{b(k)}(k)$
    When all $n_c$ bits of $s_i$ are used, the robot receives $ack \in \{0,1\}$ from the controller
    if $ack == 1$ then
        $s_i$ is appended to $\mathcal{K}_r$
    else
        $s_i$ is discarded
    end if
end loop

− − − − − − − − − − − − − − − − − − −$CONTROLLER$− − − − − − − − − − − − − − − − − − −
Initialization: Set $\mathcal{K}_c = \emptyset$.
loop    At each time step $k$ :
    the pair $(\hat{u}^D(k-1), \hat{x}(k))$ and $\hat{b}(k-1)$ are estimated using (69) and (72), respectively.
    $\hat{b}(k-1)$ is appended to the estimated codeword $\hat{c}_i$
    When $n_c$ bits of $\hat{c}_i$ are estimated, the distance $d_{\hat{c}_i}$ is computed using (74).
    if $d_{\hat{c}_i} \ll \frac{d_c-1}{2}$ then
        The codeword $\hat{c}_i$ is considered valid
        $\hat{s}_i$ is decoded from $\hat{c}_i$ and appended to $\mathcal{K}_c$; send $ack = 1$
    else
        The codeword $\hat{c}_i$ is considered invalid and discarded; send $ack = 0$
    end if
    Compute $(u^D_0(k), u^D_1(k))$ as in (71) and send it
end loop
```

adversary, the controller, and the robot can also enhance the security of the exchanged key by means of standard privacy amplification procedures, see, e.g., [Van Assche, 2006, Bennett et al., 1995].

7.3 Experimental Results

In this section, the effectiveness of the proposed key agreement protocol is verified by means of the experimental setup shown in Fig. 7.3. The setup consists of:

- A laptop where a tracking controller is implemented in Matlab.
- A Khepera IV differential-drive robot.

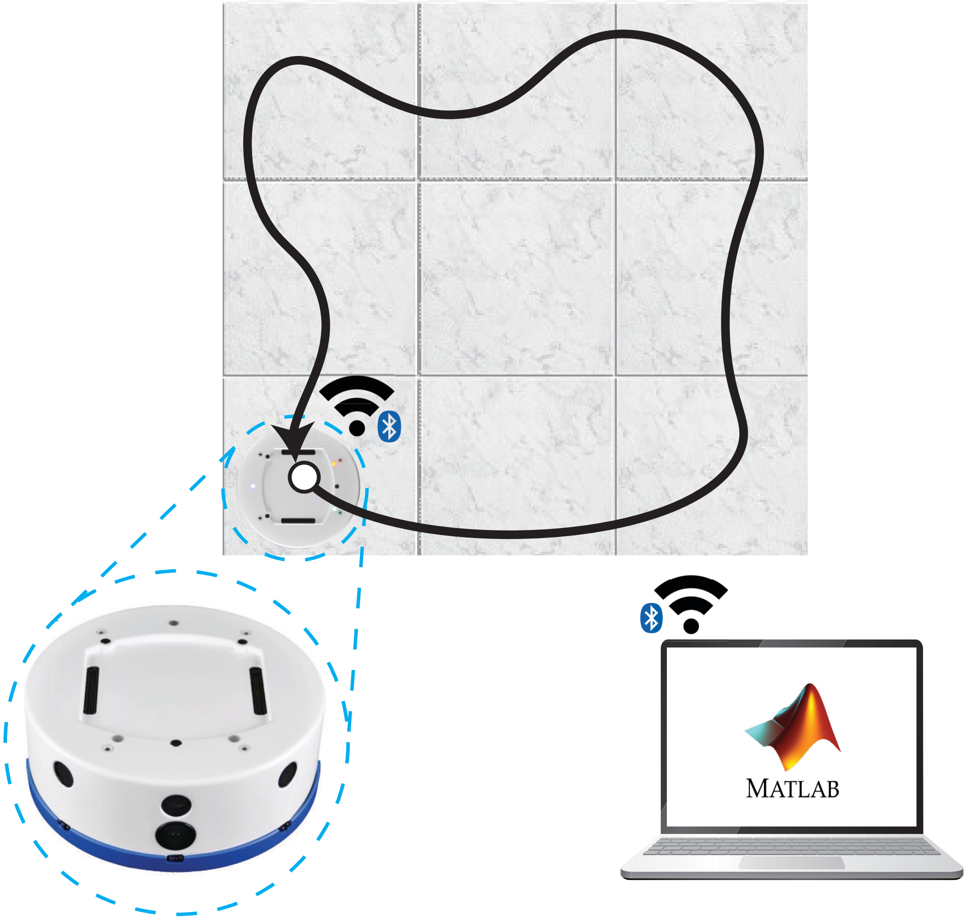


Figure 7.3: Experimental setup.

- A Bluetooth 4.0 communication channel between the robot and the laptop for the two-way exchange of data, i.e., control inputs and sensor measurements.

7.3.1 Khepera IV robot

The Khepera-IV robot, produced by K-Team, is a differential drive robot whose discrete-time kinematic model is as in (62), where, $r = 0.021\,[m]$ and $D = 0.1047\,[m]$, $\mathcal{W} = 10^{-2}I_3$, and the maximum angular velocities of the wheels is of $38\,[rad/\sec]$. On the other hand, the used measurement vector $y(k)$ consists of the wheels encoder measurements that, via odometric calculations, allow obtaining an estimation of the entire state of the robot [De Luca et al., 2001]. Consequently, the output equation (63) is modeled as $y(k) = x(k) + \xi(k)$ with $\xi(k)$ a Gaussian noise with covariance matrix $\mathcal{V} = 10^{-4}I_3$.

In the performed experiments, the robot's processing unit is equipped with a server that receives and sends, via Bluetooth, the control inputs and sensor measurements. The used sampling time is $T_s = 0.2\,[\text{sec}]$.

7.3.2 UIO, tracking Controller, and reference trajectoy

UIO: The unicycle model (65) under the control law (71) can be re-written (for compactness) as

$$x(k+1) = f(x(k), u^U(k) + \Delta_u) + \zeta(k), \tag{75}$$

with Δ_u the unknown bias of value $\pm\Delta$. Then, the extended Kalman filter with an unknown input estimation algorithm proposed in [Guo, 2018] has been used to implement the UIO module (69). For completeness, the UIO operations adapted to the considered setup, are reported in Algorithm 4, where, $P_0^x = 0_{3\times3}$, $\hat{x}_0 = [0,0,0]^T$, and A_k, B_k, G_k are the matrices characterizing the linearization $x(k+1) = A_k x(k) + B_k u^U(k) + G_k \Delta_u(k)$ of (75) along the state and input trajectories, i.e.,

$$\begin{gathered} A_k \triangleq \left.\frac{\partial f}{\partial x}\right|_{(\hat{x}_{k|k}, u^U(k))}, \quad B_k \triangleq \left.\frac{\partial f}{\partial u}\right|_{(\hat{x}_{k|k}, u^U(k))}, \\ G_k \triangleq \left.\frac{\partial f}{\partial \Delta_u}\right|_{(\hat{x}_{k|k}, u^D(k))} \end{gathered} \tag{76}$$

Consequently, $\hat{u}^D(k-1) = H^{-1}(u^U(k) + \hat{\Delta}_u(k-1))$.

Tracking controller: The robots is controlled using the nonlinear controller based on dynamic feedback linearization described in [De Luca et al., 2001, Eq. 5.18]. By denoting the reference trajectory and its first and second derivatives along the p^x and p^y axis as (p_r^x, p_r^y), $(\dot{p}_r^x, \dot{p}_r^y)$, $(\ddot{p}_r^x, \ddot{p}_r^y)$, the control law is

$$\begin{aligned} v(k) &= \ddot{p}_r^x(k) + k_p^x(p_r^x(k) - p^x(k)) + k_d^x(\dot{p}_r^x(k) - \dot{p}^x(k)) \\ \omega(k) &= \ddot{p}_r^y(k) + k_p^y(p_r^y(k) - p^y(k)) + k_d^y(\dot{p}_r^y(k) - \dot{p}^y(k)) \end{aligned} \tag{77}$$

In the performed experiments, the controller has been implemented in Matlab using $k_p^x = k_p^y = 1.10$, and $k_d^x = k_d^y = 0.80$.

Reference Trajectory: The reference signal is the square-shaped trajectory shown in Fig. 7.6.

Algorithm 4 Non-Linear Unknown Input Observer

```
**Inputs:** $u(k-1), \hat{x}_{k-1}, y(k)$
**Outputs:** $\hat{x}_k, \Delta_u(k-1)$
**Initialization**
$-----------------$ $Input\ Estimation$ $-----------------$
$\tilde{P}_{k-1} = A_{k-1}P^x_{k-1}(A_{k-1})^T + \mathcal{W}$
$\tilde{R}^*_k = \tilde{P}_{k-1} + \mathcal{V}$
$\Xi_k = (G_{k-1})^T(\tilde{R}^*_k)^{-1}$
$M_k = (\Xi_k G_{k-1})^{-1}\Xi_k$
$\hat{\Delta}_u(k-1) = M_k(y(k) - f(\hat{x}_{k-1}, u(k-1)))$
$P^a_{k-1} = M_k\tilde{R}^*_k(M_k)^T$

$-----------------$ $State\ Prediction$ $-----------------$
$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}, u(k-1) + \hat{\Delta}_u(k-1))$
$\Phi_k = (I - G_{k-1}M_k)$
$\bar{A}_{k-1} = \Phi_k A_{k-1}$
$\bar{Q}_{k-1} = \Phi_k Q_{k-1}(\Phi_k)^T + G_{k-1}M_k R_k(M_k)^T(G_{k-1})^T$
$P^x_{k|k-1} = \bar{A}_{k-1}P^x_{k-1}(\bar{A}_{k-1})^T + \bar{Q}_{k-1}$

$-----------------$ $State\ Estimation$ $-----------------$
$\Gamma_k = G_{k-1}M_k$
$\tilde{R}_k = P^x_{k|k-1} + R_k + \Gamma_k R_k + R_k(\Gamma_k)^T$
$L_k = P^x_{k|k-1} + R_k(M_k)^T(G_{k-1})^T)^T\tilde{R}^{-1}_k$
$\hat{x}_k = \hat{x}_{k|k-1} + L_k(y(k) - h_2(\hat{x}_{k|k-1}))$
$\Psi_k = I - L_k$
$P^x_k = \Psi_k P^x_{k|k-1}\Psi^T_k + L_k R_k(L_k)^T - \Psi_k$
$G_{k-1}M_k R_k(L_k)^T - L_k R_k(M_k)^T(G_{k-1})^T(\Psi_k)^T$
```

The square's vertices are $\{(0,0),(1,0),(1,1),(0,1)\}$ and the timing laws for (p^x_r, p^y_r), $(\dot{p}^x_r, \dot{p}^y_r)$, $(\ddot{p}^x_r, \ddot{p}^y_r)$ have been obtained using the built-in Matlab function *cubicpolytraj* which has been configured to travel each side of the square in 17 [sec]. In the performed experiments, the square trajectory repeats three consecutive times.

7.3.3 Perturbed control inputs and ECC configuration

Perturbed control inputs: The pair $(u^U_0(k), u^U_0(k))$ has been obtained adding a small perturbation only into the linear velocity command $v(k)$ computed as in (77), i.e., $\Delta_v > 0$ and $\Delta_\omega = 0$, see (71).

ECC configuration: A simple repetition code has been used to implement the ECC. Therefore, the string s_i consists of a single bit of $\mathcal{K}$ (i.e., $k_c = 1$), and the codewords c_i are vectors repeating s_i for n_c times. In the performed experiments, we set $n_c = 3$ and a codeword is accepted only if the number of decoding errors $d_{\hat{c}_i} = 0$.

7.3.4 Results

The proposed key-agreement protocol (Algorithm 3) has been evaluated for 10 equally spaced values of $\Delta_v \in [0.02, 0.45]$. For each Δ_v, the experiment has been repeated 10 times and with different randomly generated keys $\mathcal{K}$ of length 345 bits. The obtained results are shown in Figs. 7.4-7.7 where the shown boxplots describe the median, minimum and maximum values of each point.

Fig. 7.4 shows the percentages of accepted codewords (i.e., % of successfully decoded bits of $\mathcal{K}$) and correctly decoded/agreed bits. The number of accepted blocks (red boxplot) increases with Δ_v, which implies that the capacity of the key agreement protocol improves with the magnitude of the state shift Δ_v. Moreover, for $\Delta_v \geq 0.035$ all the accepted bits (blue boxplot) are also correct. The latter is justified by the fact that by increasing Δ_v, the distance between u_0^D and u_1^D increases until a point where estimation errors provoked by the process and measurement noises become negligible.

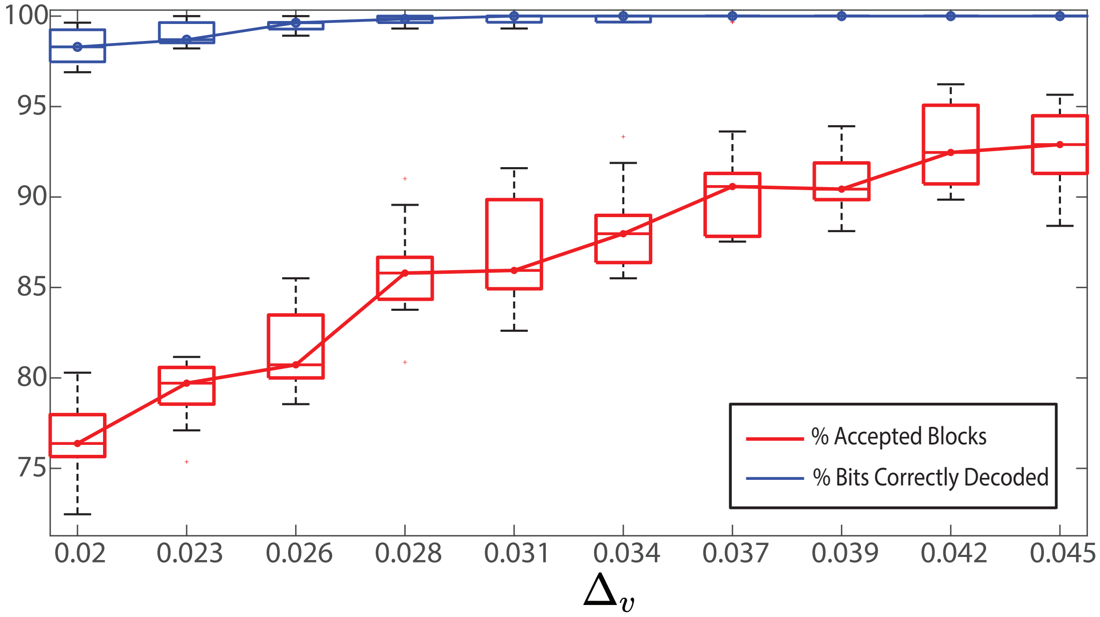


Figure 7.4: Percentage of bits accepted (red boxplot) and correctly decoded (blue boxplot) by the controller for $\Delta_v \in [0.02, 0.045]$

On the other hand, Fig. 7.5 shows the average tracking error

$$J_x = \frac{1}{N_s}\sum_{k=1}^{N_s} \|[p^x(k), p^y(k)]^T - [p_r^x(k), p_r^y(k)]^T\|_2$$

of the robot for different values of Δ_v, where N_s is the number of discrete-time steps. As expected, also the tracking error of the robot increase with Δ_v. Consequently, the latter suggests that the smallest value of Δ_v ensuring zero decoding errors (i.e., $\Delta_v = 0.035$) should be used for key-agreement. The square-shaped reference trajectory (one lap) and the robot trajectories (one lap, single experiment) in the presence (for $\Delta_v = 0.035$) and in the absence (for $\Delta_v = 0$) of the proposed key-agreement protocol are shown in Fig. 7.6. There, it is possible to appreciate how the proposed key agreement does not have a significant impact on robot reference tracking capabilities. A demo pertaining to Fig. 7.6 is available at the following weblink `https://youtu.be/9FJkQhj8sdY`.

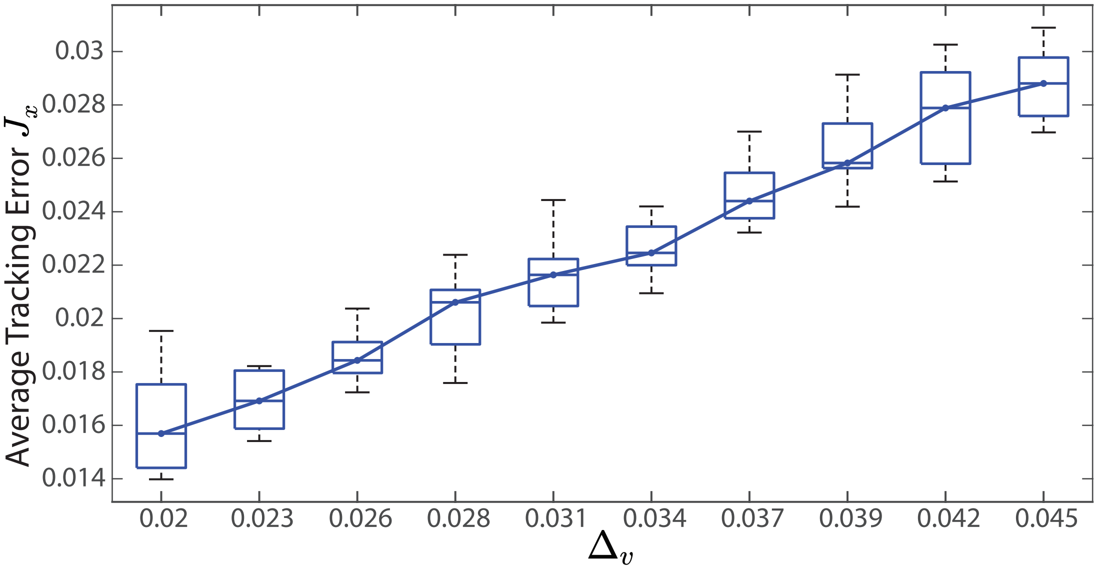


Figure 7.5: Performance index J_x for $\Delta_v \in [0.02, 0.045]$.

Finally, to test the capability of the adversary to intercept and decode the transmitted key, we have emulated an eavesdropper which has a non-perfect model knowledge $\mathcal{M}_a$. In particular, we have assumed that the attacker knows $r, d, \mathcal{W}, \mathcal{V}$ with a percentage error not superior to α. Moreover, for $\Delta_v = 0.035$, 10 equally spaced values of $\alpha \in \pm[0, 10]\%$ have been considered, and for each

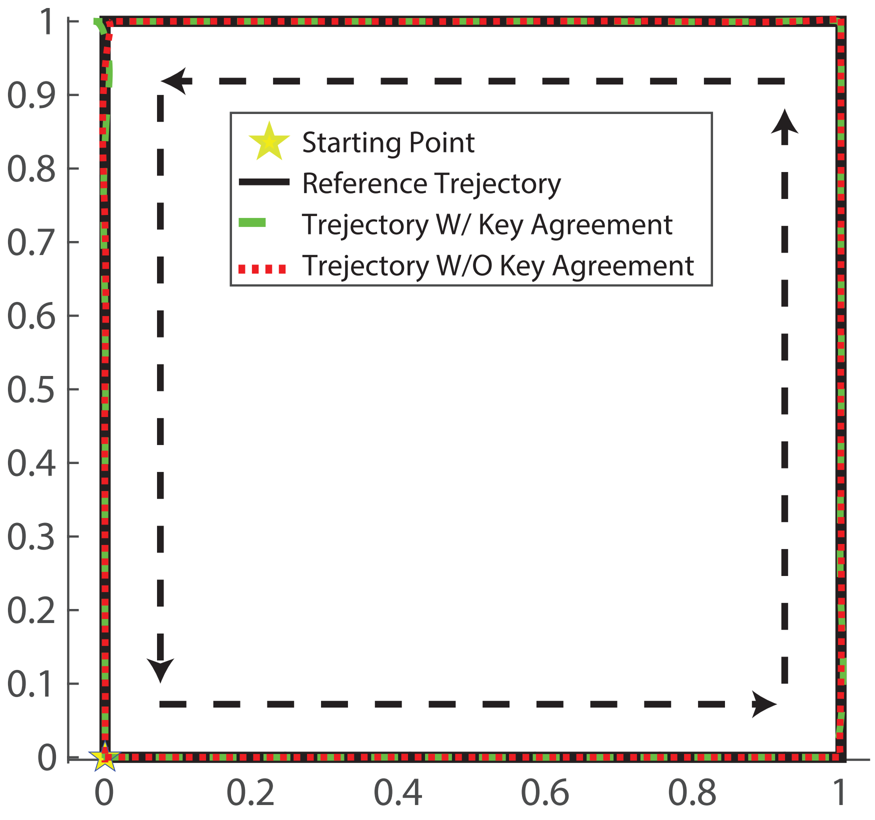


Figure 7.6: Square-shaped reference trajectory and robot's trajectory with (for $\Delta_v = 0.035$) and without the proposed key-agreement protocol.

value of α, 10 experiments have been conducted with α randomly selected in the interval $[-\alpha, \alpha]$. Fig. 7.7 reports the results of such an experiment where the y-axis shows the % bit difference between the keys estimated by the controller ($\mathcal{K}_c$) and the adversary $\mathcal{K}_a$. As expected, the adversary conceptual channel becomes worse as the model uncertainty, i.e. α, increases.

7.4 Conclusion

In this paper, we have developed a key-agreement protocol for remotely controlled mobile robots without resorting to traditional cryptographic approaches. In particular, we have leveraged the asymmetries between the controller's and adversary's knowledge about the robot model to develop a key-exchange protocol based on a non-linear unknown input observer and an error-correcting code mechanism. The proposed solution has been experimentally validated using a Khepera IV differential-drive robot, and the obtained results confirmed the effectiveness of the proposed design.

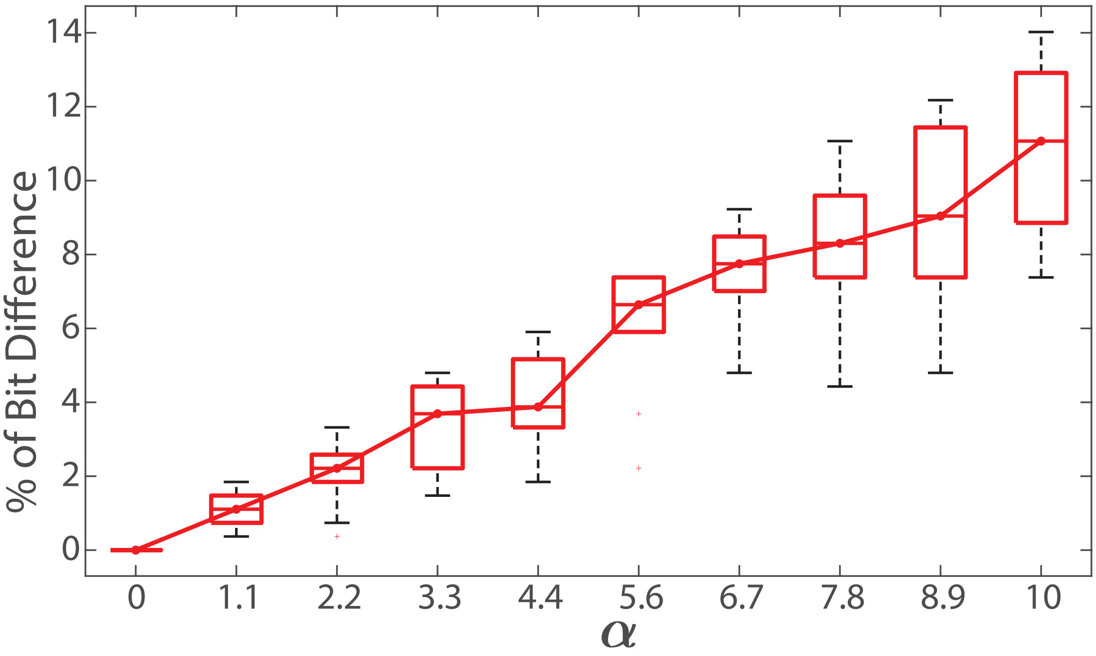


Figure 7.7: Bits difference (% disagreement) between $\mathcal{K}_c$ and $\mathcal{K}_a$ for $\Delta_v = 0.035$ and $\alpha \in \pm[0, 10]\%$.

Chapter 8

Conclusion and Future Work

This thesis deal with the problem of security and privacy of cloud-based cyber-physical systems. We investigated the advantages and drawbacks of the existing solutions in the literature and proposed different solutions to enhance the security and safety of cloud-based CPS.

- In Chapter 3, a transformation-based solution was proposed to preserve the confidentiality of sensor measurements and control input. The computational overhead introduced by the proposed solution is negligible in comparison to cryptographic methods. However, the proposed architecture provides secrecy under the assumption that the cloud has no prior knowledge about the initial state of the system and desired equilibrium points. In the future, the proposed solution can be strengthened in the presence of a more knowledgeable eavesdropper. It would be interesting to extend/enhance this scheme to be secure for situations where the adversary might be able to launch known/chosen plaintext attacks.

- To overcome the drawback of the transformation-based proposed solution, a new encrypted ST-MPC architecture was proposed in Chapter 4. In this solution, to mitigate the computational overhead introduced by the use of homomorphic encryption, we resorted to a zonotopic approximation of the used polyhedral robust one-step controllable sets. Consequently, we can reduce and control the number of operations that to be performed on the actuator. Future works can investigate the possibility of finding an admissible solution to the MPC optimization problem without using either plaintext operations on the actuator or multiple clouds.

- In Chapter 5, a security solution was developed by leveraging a trusted execution environment alongside an authenticated encryption. As the security mechanisms presented in Chapters 3 and 4, the proposed solution here guarantees the confidentiality of the data against passive attackers. However, the unique feature of this solution relies on the possibility of ensuring the integrity of the control logic against active attackers. Future works can show the effectiveness of the proposed solution in a real testbed scenario.

- In Chapter 6, by exploiting the inherent properties of the control system, particularly the small domain of the message space and the randomization utilized in homomorphic encryption schemes, we introduced a new type of attack where a malware inside the controller is able to covertly leak confidential information of the system, e.g., sensor measurement, control input or the encryption key, to an eavesdropper on the communication channels. Also, some countermeasures were proposed to nullify the introduced attacks. For future works, one may investigate other attacks that do not require compromising the random number generator. For example, Boneh et. al. [Boneh et al., 2000] showed that, under some conditions, when the length of the message is small, RSA and El-Gamal cryptosystems can be insecure. However, such attacks are probabilistic in nature and would only allow the recovery of a subset of the plaintext (measurements). Hence, it would be interesting to explore the effectiveness of such attacks in the context of encrypted control systems.

- In Chapter 7, by considering as a case study a wheeled mobile robot, a control-theoretic-based key-agreement protocol was proposed without leveraging any traditional cryptography method. The performance of the proposed protocol was evaluated through experimental results obtained with a Khepera-IV differential-drive robot. The obtained results indicated the utilized robot and the remote controller are able to agree on a random key without any significant degradation in the control performance. Future studies will be devoted develop alternative protocols capable of increasing the throughput of the key-exchange mechanism.

Bibliography

A. Abdelwahab, W. Lucia, and A. Youssef. Covert channels in cyber-physical systems. *Control Systems Letters*, 5(4):1273–1278, 2020.

R. Ahlswede and I. Csiszár. Common randomness in information theory and cryptography. i. secret sharing. *IEEE Transactions on Information Theory*, 39(4):1121–1132, 1993.

A. B. Alexandru, M. Morari, and G. J. Pappas. Cloud-based MPC with encrypted data. In *IEEE Conference on Decision and Control (CDC)*, pages 5014–5019, 2018.

D. Angeli, A. Casavola, G. Franzè, and E. Mosca. An ellipsoidal off-line mpc scheme for uncertain polytopic discrete-time systems. *Automatica*, 44(12):3113–3119, 2008.

G. Antonelli and S. Chiaverini. Linear estimation of the physical odometric parameters for differential-drive mobile robots. *Autonomous Robots*, 23(1):59–68, 2007.

V. Arun, K. Vanisree, and D. L. Reddy. Implementation of aes-gcm encryption algorithm for high performance and low power architecture using fpga. In *ISSN 1018-3639*, volume 1, pages 120–131, 2015.

P. Austrin, K.-M. Chung, M. Mahmoody, R. Pass, and K. Seth. On the impossibility of cryptography with tamperable randomness. In *Annual Cryptology Conference*, pages 462–479, 2014.

R. Baba, K. Kogiso, and M. Kishida. Detection method of controller falsification attacks against encrypted control system. In *SICE Annual Conference*, pages 244–248, 2018.

S. Bajaj and R. Sion. Trusteddb: A trusted hardware-based database with privacy and data confidentiality. *IEEE Transaction on Knowledge and Data Engineering*, 26(3):752–765, 2013.

E. Bauman, H. Wang, M. Zhang, and Z. Lin. Sgxelide: enabling enclave code secrecy via self-modification. In *Proceedings of International Symposium on Code Generation and Optimization*, pages 75–86, 2018.

H. H. Bauschke. *Projection algorithms and monotone operators*. PhD thesis, Simon Fraser University, 1996.

M. Bellare, R. Dowsley, and S. Keelveedhi. How secure is deterministic encryption? In *IACR International Workshop on Public Key Cryptography*, pages 52–73, 2015.

A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transaction on Automatic Control*, 42(3):340–349, 1997.

A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer. Generalized privacy amplification. *Transactions on Information theory*, 41(6):1915–1923, 1995.

D. P. Bertsekas and I. B. Rhodes. On the minimax reachability of target sets and target tubes. *Automatica*, 7(2):233–247, 1971.

S. Bitam and A. Mellouk. Its-cloud: Cloud computing for intelligent transportation system. In *Global Communications Conference (GLOBECOM)*, pages 2054–2059. IEEE, 2012.

F. Blanchini and S. Miani. *Set-theoretic methods in control*. Springer, 2008.

D. Boneh, A. Joux, and P. Q. Nguyen. Why textbook elgamal and rsa encryption are insecure. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 30–43. Springer, 2000.

F. Borrelli, A. Bemporad, and M. Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory*, 6(3):1–36, 2014.

F. Brasser, U. Müller, A. Dmitrienko, S. Kostiainen, K.and Capkun, and A.-R. Sadeghi. Software grand exposure: SGX cache attacks are practical. In *USENIX Workshop on Offensive Tech.*, 2017.

M. Cersullo, C. Tiriolo, G. Franzè, and W. Lucia. A detection strategy for setpoint attacks against differential-drive robots. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1035–1040, 2022.

J. H. Cheon, K. Han, H. Kim, J. Kim, and H. Shim. Need for controllers having integer coefficients in homomorphically encrypted dynamic system. In *IEEE Conference on Decision and Control (CDC)*, pages 5020–5025, 2018.

I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Tfhe: fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91, 2020.

J. Cortés, G. E. Dullerud, S. Han, J. Le Ny, S. Mitra, and G. J. Pappas. Differential privacy in control and network systems. In *IEEE Conference on Decision and Control (CDC)*, pages 4252–4272, 2016.

V. Costan and S. Devadas. Intel SGX explained. *IACR Cryptol. ePrint Arch.*, 2016(86):1–118, 2016.

M. Darup, A. Redder, and D. Quevedo. Encrypted cloud-based MPC for linear systems with input constraints. *IFAC-PapersOnLine*, 51(20):535–542, 2018.

M. Darup, G. Book, and P. Giselsson. Towards real-time ADMM for linear MPC. In *European Control Conference (ECC)*, pages 4276–4282. IEEE, 2019.

M. S. Darup. Encrypted MPC based on ADMM real-time iterations. *IFAC-PapersOnLine*, 53(2): 3508–3514, 2020.

M. S. Darup, A. B. Alexandru, D. E. Quevedo, and G. J. Pappas. Encrypted control for networked systems: An illustrative introduction and current challenges. *IEEE Control Systems Magazine*, 41(3):58–78, 2021.

S. Darup, A. Redder, I. Shames, F. Farokhi, and D. Quevedo. Towards encrypted MPC for linear constrained systems. *IEEE Control Systems Letters*, 2(2):195–200, 2017.

A. De Luca, G. Oriolo, and M. Vendittelli. Control of wheeled mobile robots: An experimental overview. *Ramsete*, pages 181–226, 2001.

S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty. A systems and control perspective of cps security. *Annual Reviews in Control*, 47:394–411, 2019.

D. Dolev and A. Yao. On the security of public key protocols. *Transaction on Information Theory*, 29(2):198–208, 1983.

L. D'Alfonso, W. Lucia, P. Muraca, and P. Pugliese. Mobile robot localization via ekf and ukf: A comparison based on real data. *Robotics and Autonomous Systems*, 74:122–127, 2015.

T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transaction on Information Theory*, 31(4):469–472, 1985.

F. Farokhi, I. Shames, and N. Batterham. Secure and private cloud-based control using semi-homomorphic encryption. *IFAC-PapersOnLine*, 49(22):163–168, 2016.

Y. Geng and M. Zhao. Homomorphic encryption technology for cloud computing. *Procedia Computer Science*, 154:73–83, 2019.

C. Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3): 97–105, 2010.

M. Ghaderi, K. Gheitasi, and W. Lucia. A blended active detection strategy for false data injection attacks in cyber-physical systems. *IEEE Transactions on Control of Network Systems*, 8(1):168–176, 2020.

O. Givehchi, J. Imtiaz, H. Trsek, and J. Jasperneite. Control-as-a-service from the cloud: A case study for using virtualized PLCs. In *Workshop on Factory Communication Systems (WFCS)*, pages 1–4. IEEE, 2014.

A. T. Gjerdrum, R. Pettersen, H. D. Johansen, and D. Johansen. Performance principles for trusted computing with intel SGX. In *International Conference on Cloud Computing and Services Science*, pages 1–18. Springer, 2017.

I. Goldberg and D. Wagner. Randomness and the netscape browser, 1996.

P. Griffioen, S. Weerakkody, and B. Sinopoli. A moving target defense for securing cyber-physical systems. *Transactions on Automatic Control*, 66(5):2016–2031, 2020.

P. Guo. *Detection and Prevention: Toward Secure Mobile Robotic Systems*. The Pennsylvania State University, 2018.

K. Ishikawa, K. Nagasawa, K. Kogiso, and K. Sawada. Experimental validation of encrypted controller implemented on raspberry pi. In *International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, pages 1–6. IEEE, 2016.

S. Jain, C. Nandhini, and R. Doriya. Ecc-based authentication scheme for cloud-based robots. *Wireless Personal Communications*, 117(2):1557–1576, 2021.

K. H. Johansson. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transaction on Control Systems Tech*, 8(3):456–465, 2000.

D. Kaplan, J. Powell, and T. Woller. AMD memory encryption. *White paper*, 2016.

J. Kim, C. Lee, H. Shim, J. H. Cheon, A. Kim, M. Kim, and Y. Song. Encrypting controller using fully homomorphic encryption for security of cyber-physical systems. *IFAC-PapersOnLine*, 49 (22):175–180, 2016.

K. Kogiso. Attack detection and prevention for encrypted control systems by application of switching-key management. In *IEEE Conference on Decision and Control (CDC)*, pages 5032–5037, 2018.

K. Kogiso and T. Fujita. Cyber-security enhancement of networked control systems using homomorphic encryption. In *IEEE Conference on Decision and Control (CDC)*, pages 6836–6843, 2015.

S. Koteshwara, A. Das, and K. K. Parhi. FPGA implementation and comparison of AES-GCM and Deoxys authenticated encryption schemes. In *IEEE International Symposium on Circuits and Systems*, pages 1–4, 2017.

B. W. Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613–615, 1973.

C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval. Key-establishment protocols for constrained cyber-physical systems. In *Security in Cyber-Physical Systems*, pages 39–65. Springer, 2021.

P. Laud and A. Pankova. New attacks against transformation-based privacy-preserving linear programming. In *International Workshop on Security and Trust Management*, pages 17–32. Springer, 2013.

J. Lee, J. Kim, and H. Shim. Zero-dynamics attack on homomorphically encrypted control system. In *International Conference on Control, Automation and Systems*, pages 385–390, 2020.

Y. Lin, F. Farokhi, I. Shames, and D. Nešić. Secure control of nonlinear systems using semi-homomorphic encryption. In *IEEE Conference on Decision and Control (CDC)*, pages 5002–5007, 2018.

W. Lucia and A. Youssef. Wyner wiretap-like encoding scheme for cyber-physical systems. *IET Cyber-Physical Systems: Theory & Applications*, 5(4):359–365, 2020.

W. Lucia and A. Youssef. A key-agreement scheme for cyber-physical systems. *Transactions on Systems, Man, and Cybernetics: Systems*, 52(8):5368–5373, 2022.

M. S. Mahmoud and Y. Xia. *Networked control systems: cloud control and secure control*. Butterworth-Heinemann, 2019.

F. N. Martins, M. Sarcinelli-Filho, and R. Carelli. A velocity-based dynamic model and its properties for differential drive mobile robots. *Journal of Intelligent & Robotic Systems*, 85(2):277–292, 2017.

U. M. Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.

D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50 (12):2967–2986, 2014.

A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography*. CRC press, 2018.

F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas. Coding schemes for securing cyber-physical systems against stealthy data injection attacks. *Transactions on Control of Network Systems*, 4(1):106–117, 2016.

C. Murguia, F. Farokhi, and I. Shames. Secure and private implementation of dynamic controllers using semihomomorphic encryption. *Transaction on Automatic Control*, 65(9):3950–3957, 2020.

A. M. Naseri, W. Lucia, M. Mannan, and A. Youssef. On securing cloud-hosted cyber-physical systems using trusted execution environments. In *International Conference on Autonomous Systems (ICAS)*, pages 1–5. IEEE, 2021.

A. M. Naseri, W. Lucia, and A. Youssef. Confidentiality attacks against encrypted control systems. *Cyber-Physical Systems*, pages 1–20, 2022a.

A. M. Naseri, W. Lucia, and A. Youssef. Encrypted cloud-based set-theoretic model predictive control. *Control Systems Letters*, 2022b.

A. M. Naseri, W. Lucia, and A. Youssef. A privacy preserving solution for cloud-enabled set-theoretic model predictive control. In *European Control Conference (ECC)*, pages 894–899. IEEE, 2022c.

H. Noura, R. Melki, A. Chehab, M. M. Mansour, and S. Martin. Efficient and secure physical encryption scheme for low-power wireless m2m devices. In *International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 1267–1272, 2018.

H. N. Noura, O. Salman, R. Couturier, and A. Chehab. Novel one round message authentication scheme for constrained iot devices. *Journal of Ambient Intelligence and Humanized Computing*, 13(1):483–499, 2022.

P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238, 1999.

N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

J. Park, A. John Park, and S. Mackay. *Practical data acquisition for instrumentation and control systems*. Newnes, 2003.

R. Parys and G. Pipeleers. Real-time proximal gradient method for linear MPC. In *European Control Conference*, pages 1142–1147. IEEE, 2018.

S. Patel, G. D. Bhatt, and J. H. Graham. Improving the cyber security of SCADA communication networks. *Communications of the ACM*, 52(7):139–142, 2009.

S. Rakovic, E. Kerrigan, K. Kouramas, and D. Mayne. Invariant approximations of the minimal robust positively invariant set. *Transaction on Automatic Control*, 50(3):406–410, 2005.

D. B. Rawat, T. White, M. S. Parwez, C. Bajracharya, and M. Song. Evaluating secrecy outage of physical layer security in large-scale mimo wireless communications for cyber-physical systems. *Internet of Things Journal*, 4(6):1987–1993, 2017.

J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, 2017.

B. D. Ripley. Thoughts on pseudorandom number generators. *Journal of Computational and Applied Mathematics*, 31(1):153–163, 1990.

N. Schlüter and M. S. Darup. Encrypted explicit MPC based on two-party computation and convex controller decomposition. In *IEEE Conference on Decision and Control (CDC)*, pages 5469–5476, 2020.

Z. Shan, K. Ren, M. Blanton, and C. Wang. Practical secure computation outsourcing: A survey. *Computing Surveys (CSUR)*, 51(2):1–40, 2018.

C. Shepherd, G. Arfaoui, I. Gurulian, R. P. Lee, K. Markantonakis, D. Akram, R. N.and Sauveron, and E. Conchon. Secure and trusted execution: Past, present, and future-a critical review in the context of the internet of things and cyber-physical systems. In *IEEE Trustcom/BigDataSE/ISPA*, pages 168–177, 2016.

X. Shi, S. Shi, M. Wang, J. Kaunisto, and C. Qian. On-device iot certificate revocation checking with small memory and low latency. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1118–1134, 2021.

R. S. Smith. Covert misappropriation of networked control systems: Presenting a feedback structure. *Control Systems Magazine*, 35(1):82–92, 2015.

H. H. Sohrab. *Basic real analysis*, volume 231. Springer, 2003.

S. Sokolov, V. Kamenskij, A. Novikov, and V. Ivetić. How to increase the analog-to-digital converter speed in optoelectronic systems of the seed quality rapid analyzer. *Inventions*, 4(4):61, 2019.

A. K. Sutrala, M. S. Obaidat, S. Saha, A. K. Das, M. Alazab, and Y. Park. Authenticated key agreement scheme with user anonymity and untraceability for 5g-enabled softwarized industrial cyber-physical systems. *Transactions on Intelligent Transportation Systems*, 23(3):2316–2330, 2021.

A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. A secure control framework for resource-limited adversaries. *Automatica*, 51:135–148, 2015.

K. Teranishi and K. Kogiso. Control-theoretic approach to malleability cancellation by attacked signal normalization. *IFAC-PapersOnLine*, 52(20):297–302, 2019.

K. Teranishi, N. Shimada, and K. Kogiso. Stability analysis and dynamic quantizer for controller encryption. In *Conference on Decision and Control (CDC)*, pages 7184–7189. IEEE, 2019.

J. Tran, F. Farokhi, M. Cantoni, and I. Shames. Digital implementation of homomorphically encrypted feedback control for cyber-physical systems. *github.io*, 2019.

J. Tran, F. Farokhi, M. Cantoni, and I. Shames. Implementing homomorphic encryption based secure feedback control. *Control Engineering Practice*, 97, 2020.

G. Van Assche. *Quantum cryptography and secret-key distillation*. Cambridge University Press, 2006.

J. Varia, S. Mathew, and et. al. Overview of amazon web services. *Amazon Web Services*, 105, 2014.

C. Wang, K. Ren, and J. Wang. Secure and practical outsourcing of linear programming in cloud computing. In *IEEE Infocom*, pages 820–828, 2011.

P. Weeraddana and C. Fischione. On the privacy of optimization. *IFAC-PapersOnLine*, 50(1): 9502–9508, 2017.

A. D. Wyner. The wire-tap channel. *Bell System Technical Journal*, 54(8):1355–1387, 1975.

Z. Xu and Q. Zhu. Secure and resilient control design for cloud-enabled networked control systems. In *ACM Workshop on Cyber-Physical Systems-Security and/or Privacy*, pages 31–42, 2015.

J.-P. A. Yaacoub, H. N. Noura, O. Salman, and A. Chehab. Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations. *International Journal of Information Security*, pages 1–44, 2021.

K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester. A2: Analog malicious hardware. In *IEEE Symposium on Security and Privacy*, pages 18–37, 2016.

L. Yang and N. Ozay. Scalable zonotopic under-approximation of backward reachable sets for uncertain linear systems. *IEEE Control Systems Letters*, 6:1555–1560, 2021.

X. Yi, R. Paulet, and E. Bertino. Homomorphic encryption. In *Homomorphic Encryption and Applications*, pages 27–46. Springer, 2014.

T. Zaslavsky. *Facing up to arrangements: Face-count formulas for partitions of space by hyperplanes*, volume 154. American Mathematical Soc., 1975.

Y. Zhang, Y. Xiang, and X. Huang. A cross-layer key establishment model for wireless devices in cyber-physical systems. In *ACM Workshop on Cyber-Physical System Security*, pages 43–53, 2017.

L. Zhou, V. Varadharajan, and M. Hitchens. Achieving secure role-based access control on encrypted data in cloud storage. *IEEE Transaction on Information Forensics and Security*, 8(12): 1947–1960, 2013.